CANADIAN THESES ON MICROFICHE

THÈSES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

Canada

NL 339 (r. 86/08)

National Library of Canada · Bibliothèque nationale du Canada

Ottawa, Canada
K1A 0N4

TC –

ISBN 0-315-21281-0

CANADIAN THESES ON MICROFICHE SERVICE – SERVICE DES THÈSES CANADIENNES SUR MICROFICHE

PERMISION TO MICROFILM – AUTORISATION DE MICROFILMER

• Please print or type – Écrire en lettres moulées ou dactylographier

AUTHOR – AUTEUR

Full Name of Author – Nom complet de l'auteur

ILLEPERUMAGE ARIYAWANSA PERERA

Date of Birth – Date de naissance

JAN 23/50

Canadian Citizen – Citoyen canadien

☐ Yes / Oui ☑ No / Non

Country of Birth – Lieu de naissance

SRI LANKA

Permanent Address – Résidence fixe

RAMUKKANA
BANDARAGAMA
SRI LANKA

THESIS – THÈSE

Title of Thesis – Titre de la thèse

SELECTING TEST DATA FOR THE DOMAINS
TESTING STRATEGY

Degree for which thesis was presented
Grade pour lequel cette thèse fut présentée

M.Sc

Year this degree conferred
Année d'obtention de ce grade

1985

University – Université

UNIV. OF ALBERTA

Name of Supervisor – Nom du directeur de thèse

LEE J. WHITE

AUTHORIZATION – AUTORISATION

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to **microfilm** this thesis and to **lend** or **sell copies of the film.**

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de **microfilmer** cette thèse et de **prêter** ou de **vendre des exemplaires du film.**

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

ATTACH FORM TO THESIS – VEUILLEZ JOINDRE CE FORMULAIRE À LA THÈSE

Signature

Date

April 24/85

NL-91 (r. 84/03)

Canada

THE UNIVERSITY OF ALBERTA

SELECTING TEST DATA FOR THE DOMAIN TESTING STRATEGY

by

I. ARIYAWANSA PERERA

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

SPRING 1985

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR I. ARIYAWANSA PERERA

TITLE OF THESIS SELECTING TEST DATA FOR THE DOMAIN TESTING STRATEGY

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED SPRING 1985

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) .....

PERMANENT ADDRESS:

Ramukkana

Bandaragama

Sri Lanka

DATED April 24 1985

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled SELECTING TEST DATA FOR THE DOMAIN TESTING STRATEGY submitted by I. ARIYAWANSA PERERA in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE.

Lee J. White
Supervisor

Clifford A. Adver

J. E. Lewis

L. Soltz

Date April 23, 1985

ABSTRACT

The Domain Testing Strategy is a testing method designed to uncover control flow errors in a selected path of a computer program. The objective of this thesis is to study and develop possible methods to select test data for domain testing with an acceptable error bound. In evaluating these methods the error bound, the number of required test points and the time required to test a domain are all considered. Three proposed test data selection methods are determined to be intractable, since they have a nonpolynomial time complexity. This thesis proposes a new method which can be done in polynomial time.

Difficulties in applying domain testing in a discrete space or in a space in which numerical values can only be represented with finite resolution are studied. The method developed for continuous space is adapted to select discrete test points without much additional effort.

ACKNOWLEDGEMENTS

I wish to express my gratitude to my supervisor Dr. Lee J. White for his guidance, assistance and patience throughout the course of this work.

I am also grateful to Dr. J. E. Lews for his help in solving many mathematical problems of this thesis.

Further thanks are due to the other members of my examining committee, Drs. C. A. Addison and S. Cabay for their valuble comments.

Finally, I am deeply grateful to my wife Chandanie for her continual love and support.

Table of Contents

Chapter		Page
1.	INTRODUCTION	1
2.	DOMAIN TESTING STRATEGY	5
	2.1 Terminology	5
	2.2 The Domain Strategy	8
	2.3 Error Analysis of Domain Testing.	15
	2.4 An Alternative Method to Select Test Points.	20
3.	DOMAIN TESTING IN HIGHER DIMENSIONS	25
	3.1 The Nx1 Strategy.	26
	3.2 Error Analysis in N-Dimensions.	27
	3.2.1 BSE as the Error Measure in N-dimensions.	27
	3.2.2 An Alternative Error Measure in N-dimensions.	30
	3.2.3 A Sufficient Condition for Finite Error in Higher Dimensions.	38
	3.3 Other Strategies.	42
	3.3.1 The NxN Strategy.	42
	3.3.2 The VxV Strategy.	44
	3.4 Evaluation of the Strategies.	45
	3.5 An Alternate Strategy.	49
	3.6 An Upper Bound of the MAD of the Scattering Method in 3-Dimensions	53
	3.7 Summary	55
4.	DOMAIN TESTING IN DISCRETE INPUT SPACE	58
	4.1 Discrete Input Space.	58
	4.2 Selecting Test Points in a Discrete Space.	59
	4.2.1 ON-Points.	59
	4.2.2 OFF-Points.	69

4.3 Fundamental Limitations in a Discrete Space. 70

5. CONCLUSIONS AND FUTURE WORK 75

5.1 Overview 75

5.2 Contributions to Domain Testing 77

5.3 Future Work 78

REFERENCES 80

APPENDIX A A Geometrical Interpretation of the Angle Between Two Hyperplanes 83

APPENDIX B An Upper Bound for the MAD in 3-Dimensions ...85

APPENDIX C A Sufficient Condition for an Acute MAD in 3-Dimensional Discrete Space 86

List of Figures

Figure		Page
2.1	Border Segments of a Path Domain	11
2.2	Domain Testing in 2-Dimensions	14
2.3	Four Types of Border Shifts.	16
2.4	Unbounded BSE for 2x1 Strategy	18
2.5	Limiting Positions of the Potential Actual Border	18
2.6	Subpath Domains	22
3.1	Undetectable Potential Border Shifts	29
3.2	Two Possible Situations of Unbounded Error	33
3.3	A Geometrical Interpretation of the MAD	35
3.4	Optimizing the MAD for the 3x1 Strategy	35
3.5	Selecting Vertices as ON-Points	37
3.6	Nearly Parallel Adjacent Edges	39
3.7	Calculating the MAD in 3-Dimensions	43
3.8	Vertices not Adjacent to the Border Segment	47
3.9	Scattering Method in 3-Dimensions	52
3.10	Parallelogram Generated by the Scattering Method in 3-Dimensions	54
3.11	Worst Case for ON-Points	54
4.1	Correct Borders in 2-Dimensional Discrete Space	60
4.2	Selection of a Discrete ON-Point in the Semicircle Close to the Vertex E	63
4.3	Position of a Discrete ON-Point Close to the Vertex E	64
4.4	Parallel Hyperplane Segment at Distance $\sqrt{N}\Delta/2$	68

Figure Page

4.5 Examples for Domains with an Insufficient Width. 71

CHAPTER 1

INTRODUCTION

The phrase "reliable software" has two distinct shades of meaning [18]. On one hand it implies *correctness*. A piece of software is correct if it meets its functional specifications. This implies that as long as the inputs satisfy the specifications, the software will produce the desired outputs. On the other hand, reliable software is *robust*, or fault tolerant, if it can be expected to deliver a certain minimum level of service even when faced with an unexpected or hostile environment. Thus, a piece of software is reliable if it is both consistent with respect to the stated specifications and able to withstand unexpected demands.

Having written a program, how do programmers know that it is the program they meant to write? In other words, how do they convince themselves, let alone other people, that they have produced the correct program? One process they may follow in doing this has come to be known as software validation.

Program testing is the most widely spread technique of software validation. It is concerned with analyzing a program by evaluating its response to a selected set of input data. To test a program, a test criterion is needed to guide the selection or generation of test data. White and Cohen [15, 16] have developed a method called the *domain*

testing strategy, which focuses on the detection of control flow errors.

As defined by Howden [7] a computer program may have two types of errors called 'computation errors' and 'domain errors'. A *computation error* occurs when a specific input follows the correct path, but an error in some assignment statement causes the wrong function to be computed for one or more output variables. A program contains a *domain error* when a specific input causes the execution of a wrong path due to an error in the control flow of the program. This type of error can occur due to an incorrect predicate expression or due to an error in some assignment statement which affects a subsequent predicate. Thus, an error in an assignment statement may cause both types of error. This error classification is by no means exhaustive. Clearly there exist other types of errors such as syntax errors, errors in input and output, etc.

The domain testing strategy, which has been designed to detect domain errors or to obtain confidence in the correctness of the control flow of a path, falls within a class of strategies called path analysis testing. These type of testing processes are conducted in two steps [1,6,11].

1. Select a set of paths for testing [17,18,20].
2. Perform testing along the chosen path.

The domain testing strategy deals with the second step only. It finds a set of test data for a selected path and evaluates the output generated by running the program on

selected test data to determine the correctness of the control flow of the path.

No test criterion can guarantee the complete correctness of a program unless it selects the entire program domain as the set of test data [13]. Such exhaustive testing is clearly impractical. The practicability of a testing method depends on two conflicting attributes, the degree of reliability guaranteed by the test and the cost of the test.

On the basis of geometrical arguments White and Cohen developed a method to select test data for domain testing of a program subject to certain limitations. Several alternative methods have been subsequently developed. The objective of this research is to analyze these methods and to improve domain testing with respect to the correctness and the cost.

The organization of the thesis is as follows. In Chapter 2 the Domain Testing Strategy is introduced as it was developed for the 2-dimensional input space. It also develops a new criterion to select test points to obtain better accuracy. Chapter 3 is mainly concerned with the application of domain testing in higher dimensional continuous input spaces. A new error measure which simplifies the analysis in N-dimensions is introduced. Some proposed test point selection strategies are discussed to examine their applicability. Considering the time complexity and the required number of test points of a test, a new

testing method is developed.

The application of the domain testing strategy in discrete input space is considered in Chapter 4. The new testing method developed for the continuous space can be easily adapted to an N-dimensional discrete input space. It is more difficult to find the required test points in a discrete input space than in a continuous space. This situation limits the application of domain testing in a discrete input space. Possibilities of characterizing these pathological situations for which domain testing cannot be applied are examined. Chapter 5 will provide conclusions and future research problems arising from this work.

CHAPTER 2

DOMAIN TESTING STRATEGY

2.1 Terminology

The variables used in a computer program can be divided into two major classes. If the value of the variable is assigned through an input statement, i.e. a READ statement, it is classified as an *input variable* and all the other variables are called *program variables*. The values of the program variables are given by assignment statements. The *input space* of a program with input variables $v_1, v_2, \ldots, v_N$ is the set of N-tuples $(a_1, a_2, \ldots, a_N)$ where a_i is some possible value of the variable v_i for $i=1,2,\ldots,N$. Then the dimension of the input space is N, the number of input variables.

A *path* through a program can be defined as a set of sequentially arranged program statements, including the initial and final statements, according to some possible flow of execution. It should be noted that two paths which differ only in the number of times a particular loop in the program is executed are considered different paths. Thus, the number of paths of a program can be infinite. Associated with each path is the *path domain*, the set of input values which cause the execution of the path, and the *path computation*, the function that is computed by the execution of the statements along the path. A path may be *infeasible* in the sense that no input point exists which will cause the

path to be executed, i.e., it has an empty path domain.

Each conditional branch which is taken along a path is associated with an iteration loop or a predicate which evaluates to true or false, and its value determines which outcome of the branch will be followed. A predicate is a logical combination of one or more relational expressions combined by the relational operators OR and AND. For the purpose of this study only the predicates with arithmetic relational expressions are considered. If a predicate has one or more relational expressions with boolean operators, then it is a *compound predicate*. A *simple predicate* consists of just a single relational expression.

In general, a predicate will be expressed in terms of both program variables and input variables. However, to determine a path domain we must work with constraints in terms of input variables. Corresponding to a particular path, we can obtain an equivalent constraint of a predicate in terms of input variables by symbolic evaluation. Such an expression of a predicate is called a *predicate interpretation*. A single predicate can appear on many different paths. Since each of these paths will, in general, consist of a different sequence of assignment statements, a single predicate can have many different interpretations. The following program segment provides example predicates and interpretations.

```
READ A,B
IF A > B
    THEN C = B + 1
    ELSE C = B - 1
ENDIF
D = 2*A + B
IF C ≤ 0
    THEN E = 0
    ELSE
        DO I = 1,B
            E = E + 2*I
        ENDDO
ENDIF
IF D = 2
    THEN F = E + A
    ELSE F = E - A
ENDIF
WRITE F
```

In the first predicate, A > B, both A and B are input variables, so there is only one interpretation. The second predicate, C ≤ 0, will have two interpretations depending on which branch was taken in the first IF construct. For paths on which the THEN C = B + 1 clause is executed, the interpretation is B + 1 ≤ 0 or equivalently B ≤ -1. When the ELSE C = B - 1 branch is taken, the interpretation is B - 1 ≤ 0, or equivalently B ≤ 1. According to the number of

times the DO loop is executed, the third predicate D = 2 can appear in several paths. Nevertheless, it only has one interpretation, 2*A + B = 2, since D is assigned the value 2*A + B in the same statement in each path.

An interpretation of a simple predicate is *linear* in the input variables if it is of the form:

$$c_1v_1+c_2v_2+ \ldots\ldots + c_Nv_N \text{ ROP } k$$

where $v_1,\ldots,v_N$ are input variables, $c_1,\ldots,c_N$ and k are constants. ROP represents one of the relational operators ($<,>,=,\leq,\geq,\neq$). An interpretation of compound predicate is linear in the input variables when each of its components is linear.

2.2 The Domain Strategy

The White and Cohen *domain testing strategy* [15, 16] has been designed to detect domain errors or to obtain confidence in the correctness of a path domain. It does not deal with selecting suitable paths but tests for domain errors in a selected path. There are limitations inherent to any testing method, and these also constrain the domain testing strategy [15, 16]. One such limitation we shall define as *coincidental correctness*, which can occur when a specific test point follows an incorrect path, and yet the output variables coincidentally are the same as if that test point were to follow a correct path. This test point would then be of no assistance in the detection of a domain error which could cause the control flow change. Another basic

limitation is called a *missing path error*, in which a required predicate does not appear in the given program to be tested. This generally occurs when a programmer forgets that certain input values are to be treated as separate or special cases.

The domain testing strategy is initially developed, explained and validated in detail under the following assumptions:

1. Coincidental correctness does not occur for any test case. If correct output results are produced, we can assume that the test point is in the correct path domain.
2. A missing path error is not associated with the path being tested.
3. Each border is produced by a simple predicate.
4. The path corresponding to each adjacent domain computes a different function from the path being tested.
5. The given border is linear, and if it is incorrect the actual border is also linear.
6. The input space is continuous rather than discrete.
7. The input space is 2-dimensional, corresponding to a program which reads, at most, two input variables.

Any program which satisfies the first six assumptions will be referred to as a *linearly domained program*.

Assumptions (1) and (2) have been shown [15] to be inherent to the testing process for the path oriented methods, and cannot be entirely eliminated. Assumptions (3) through (7) are for convenience in the initial development and relaxed afterwards. This research specifically analyzes the application of domain testing in both continuous and discrete spaces with an arbitrary number of input variables.

An input point belongs to a given domain if it satisfies the *path conditions* defined by the interpretations of all the predicates along the path and the minimum and maximum values of the input variables. The boundary of the domain, therefore, is determined by these path conditions. Under the above set of assumptions a path domain is a convex polygon, a line segment, a point or a null set. The boundary of a polygonal domain consists of line segments which are the sides of the polygon. Each line segment is a part of a straight line which corresponds to a single path condition. The extreme points of such a line segment is determined by two or more other path conditions. For an example, consider the path of a program with four path predicates:

P_1: $5y-4x \leq 10$

P_2: $x \geq 3$

P_3: $y+x \geq 9$

P_4: $2y \geq 5$

Two input variables x and y are bounded by $0 \leq x, y \leq 10$.

The quadrilateral EBHG in Figure 2.1 depicts the given domain. The boundary of the domain consists of four line

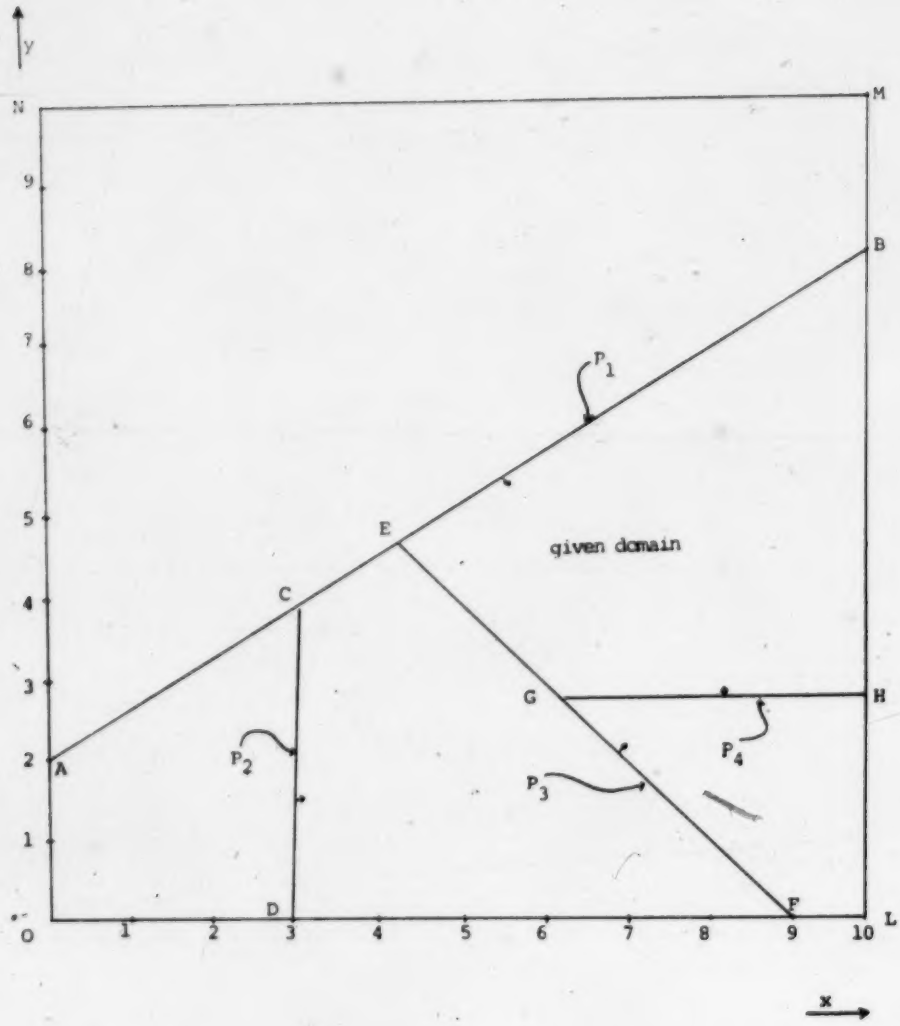


Figure 2.1 Border Segments of a Path Domain

segments EB,BH,HG and EG. The line segment EB corresponds to the predicate P_1 and the extreme points E and B are determined by the predicates P_3 and the upper bound of the variable x. The line segment BH is defined by the upper bound condition $x \leq 10$. Two extreme points B and H of that line segment are determined by the predicates P_1 and P_4 respectively. The conditions which determine the minimum and maximum values of the variable y, the minimum value of the variable x and the predicate P_2 do not contribute to the boundary of the given domain.

A line segment of the boundary which corresponds to a path predicate is called a *border segment* of the path domain. A *closed border segment* is a part of the domain and is formed by a predicate with $\leq$, $\geq$ or = operators. An *open border segment* is a part of the domain boundary but does not constitute part of the domain, and is formed by <, > and $\neq$ operators. The domain testing strategy selects test data for each border segment of the domain being tested. A domain error is a shift of a border segment due to an error in the corresponding predicate or an error in an assignment statement which affects that predicate. Therefore the points close to the border are the most sensitive to domain errors [1]. Using this fact, test data are selected on or very close to the border.

Two types of test data, called *ON-points* and *OFF-points*, are selected to test a border segment of a domain to be tested. An ON-point is selected on or close to

the border segment such that it satisfies all the path conditions. An OFF-point is selected at a small distance ϵ from the border segment satisfying all the path conditions except the condition which corresponds to the given border segment.

If the program produces correct output for all the test data, it can be concluded that the actual border should lie between ON-points and OFF-points. That is, the actual border should intersect all the ON-OFF line segments. A straight line in a 2-dimensional space is uniquely determined by any two points. So, a actual border segment can be identified if we can find any two points on the border segment. Since the actual border must intersect every ON-OFF line, we need at least two ON-OFF line segments. White and Cohen [15, 16] achieve this by testing two ON-points and a single OFF-point whose projection on the given border is a convex combination of the projection points of the ON-points. This strategy is called 2x1 strategy by Clarke et al.[2].

Figure 2.2 illustrates the basic idea of the strategy. To test the closed border segment EF of the domain D, two points A and B are selected as ON-points. Both points satisfy all the path conditions. C is the OFF-point and it satisfies all the path conditions except the condition that correspond to EF. Also C', which is the projection of C on EF, is a point between A' and B', which are the projection points of A and B on EF. If the given border segment EF is incorrect and the actual border intersects both line

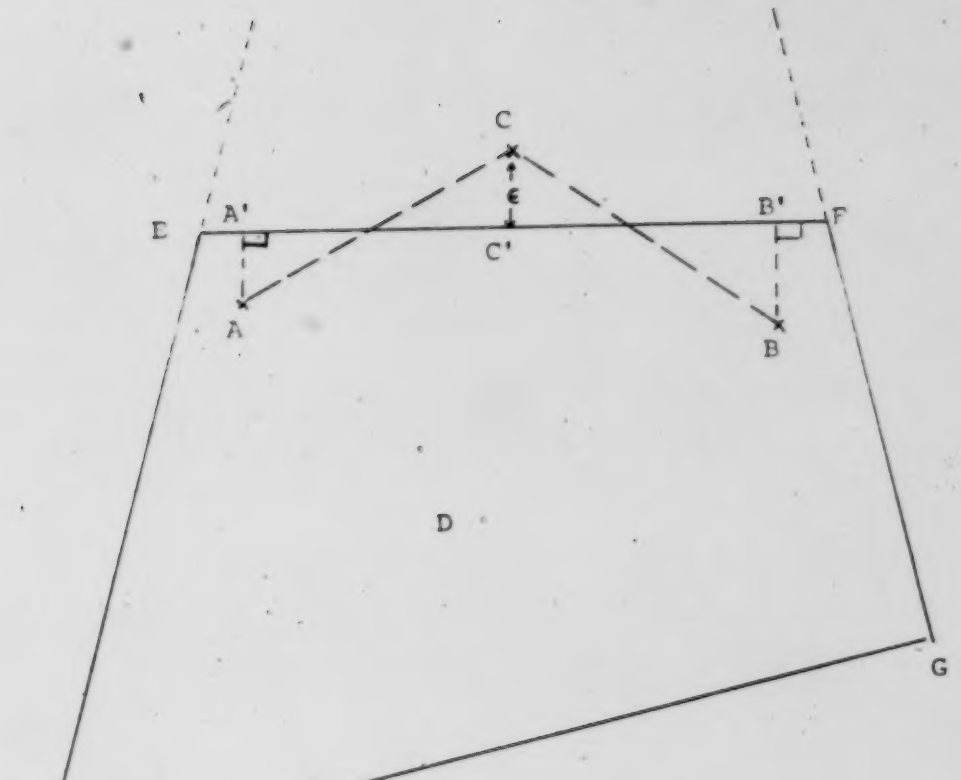


Figure 2.2 Domain Testing in 2-Dimensions

segments AC and BC then the error cannot be detected by these three points. Any border shift beyond this level will be detected by producing wrong output for at least one test point.

Figure 2.3 shows the four general types of border shifts. In Figure 2.3(a) the actual border intersects both ON-OFF line segments AC and BC. Since all three test points are in correct domains with respect to the actual border, the error will not be detected. In Figure 2.3(b) the border shift has reduced the domain D_1, causing point C to be in domain D_2 instead of D_1. It will yield an incorrect output while A and B which are in the correct domain produce the correct outputs. The border shift in Figure 2.3(c) has increased the domain D_1, causing the points A and B to be in the domain D_1 instead of D_2. The border shift, therefore, will be detected by the incorrect outputs of A and B. Finally the border shift in Figure 2.3(d) will be detected, for it causes the point B to be in a wrong domain. Points A and C will yield the correct outputs since they remain in correct domains. Thus, the border shifts shown in Figure 2.3 (b), (c) and (d) cause to produce at least one incorrect output and will be detected by the given test points.

2.3 Error Analysis of Domain Testing.

The correct output for all the test data points in domain testing does not guarantee that the given border is correct, as there is still a possibility of an undetected

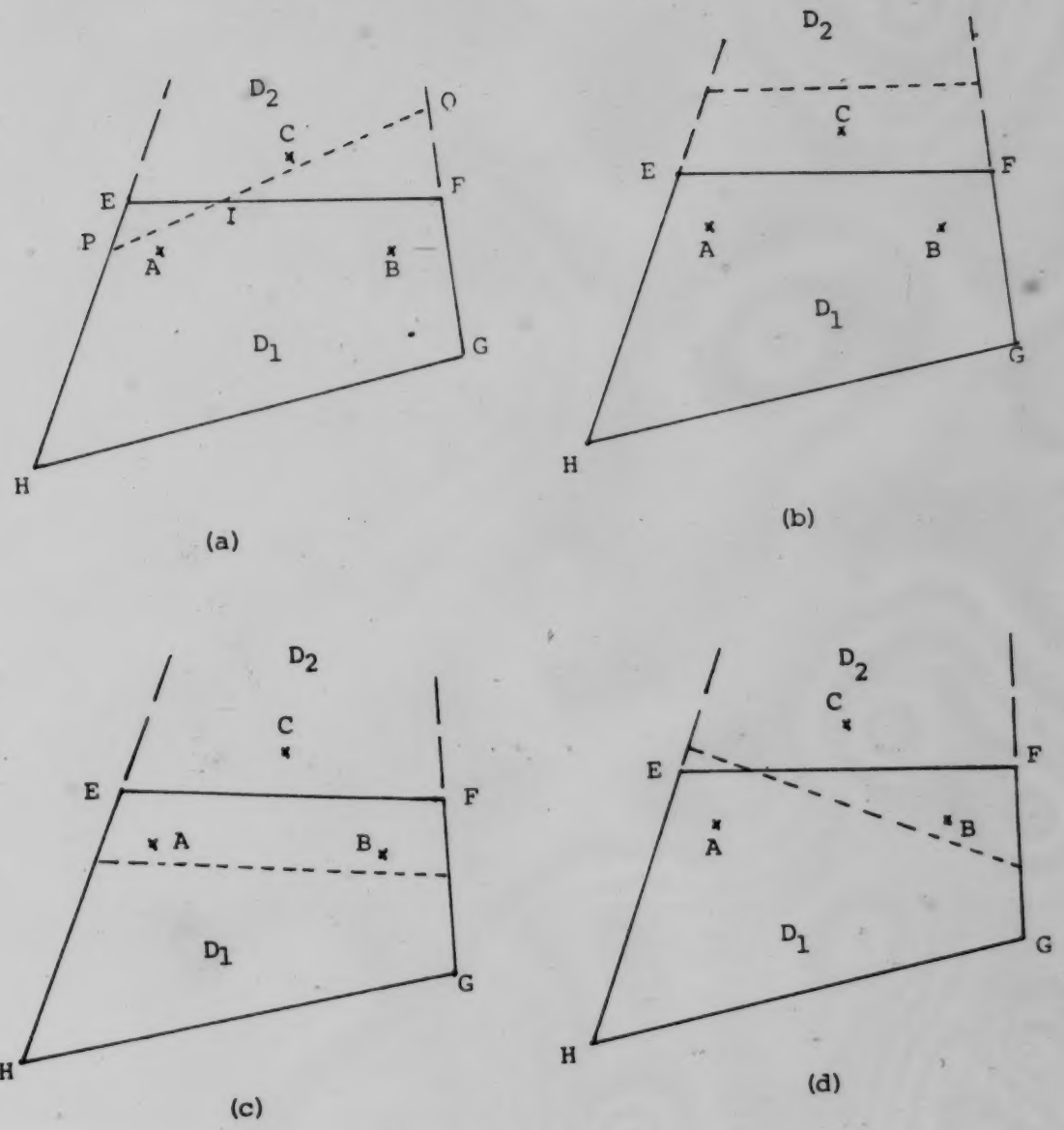


Figure 2.3 Four Types of Border Shifts.

error. For instance, all the input points in the triangles EIP and QIF in Figure 2.3(a) lie in a wrong domain. The given 2x1 test points A, B and C will not detect this error.

To analyze the error of a testing method, the first issue to be addressed is to obtain a measure of the error. The error analysis of the domain testing strategy is done on the basis of the maximum area of the input space which may belong to a wrong domain due to an undetected error of the border segment being tested [14]. This error measure is called the *Border Shift Error (BSE)* [2]. This is the maximum area between the given border, given adjacent borders and a possible actual border. When a possible actual border does not intersect all the given adjacent borders, this area can be unbounded as shown in Figure 2.4. It has been shown [14] that an unbounded error occurs when an adjacent border is nearly parallel to the given border, i.e., the angle between an adjacent border and the given border is either near 0° or near 180°.

The BSE, when it is bounded, is determined by a potential position of the actual border which maximizes the area in a wrong domain. Since it is difficult to find such a line segment, we consider four possible positions which limit the undetectable border shifts that can happen due to an error in a coefficient or the constant term of the predicate interpretation. As shown in Figure 2.5, to test the border segment EF, points A and B are used as the ON-points and C as the OFF-point. Then the four limiting

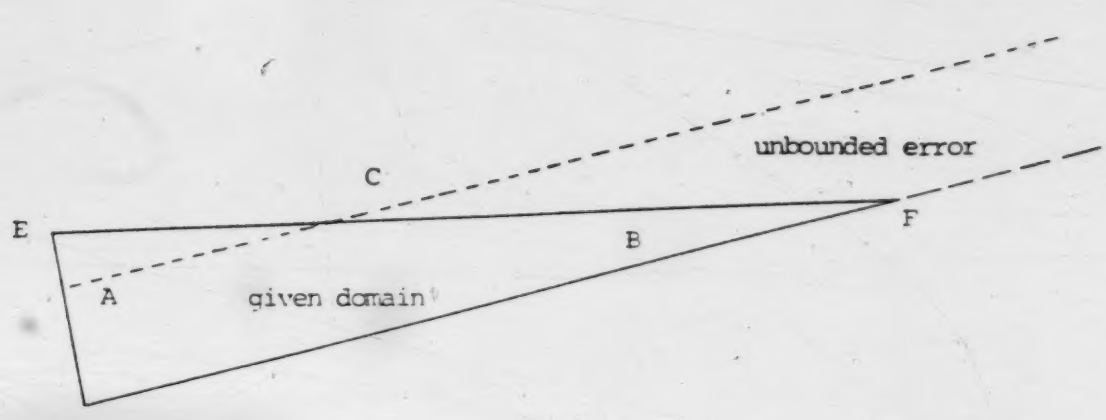


Figure 2.4 Unbounded BSE for 2x1 Strategy

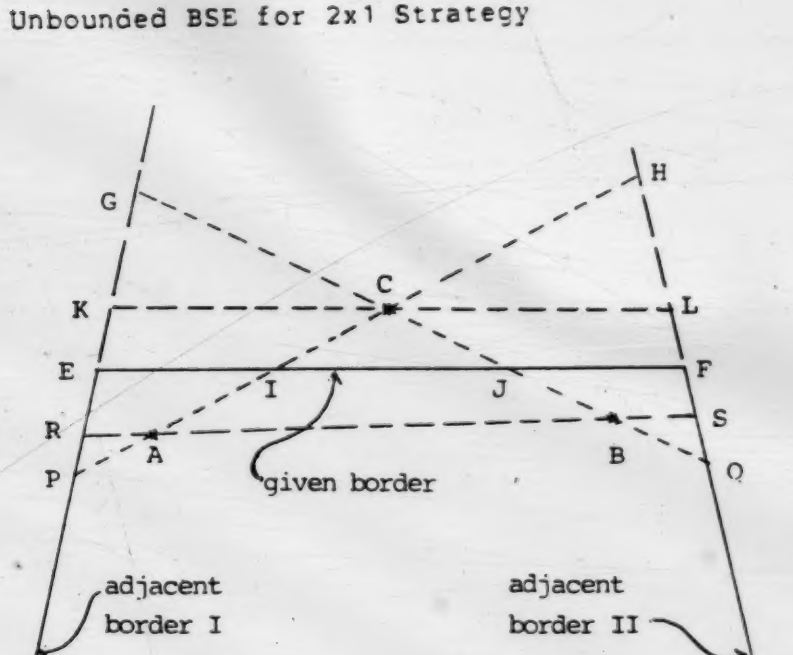


Figure 2.5 Limiting Positions of the Potential Correct Border

positions are:

1. the line segment PH, determined by the test points A and C; then the erroneous space is composed by two triangles PIE and HIF;
2. the line segment QG, determined by the points B and C; the triangles QFJ and GJE form the erroneous space;
3. the line KL, determined by the point C and parallel to EF, causes the trapezoidal area EFLK to be in a wrong domain;
4. the line RS, determined by A and B; then the quadrilateral area ERSF is in a wrong domain.

One drawback of the BSE as the error measure criterion for the domain testing is the assumption that the given adjacent borders can be extended infinitely which may not be valid. There can be redundant predicates which limit the extension of the given adjacent borders. Moreover, the input space is not unbounded. It is bounded according to the format used to read the input data and the minimum and maximum values of a variable which can be represented within a computer.

The BSE is used comparatively to evaluate a test. If one domain testing method always provides a smaller BSE than another method, we consider the former to be more reliable, hence better.

White et al. [14] have developed some general guidelines to select test points to obtain a good 2x1 test with small BSE.

1. The ON-points should be selected at, or as close as possible to, the vertices of the given border segment.
2. The distance between the OFF-point and the given border should be very small and the OFF-point is selected close to the midpoint of the border segment.

Clarke et al. [2] have proposed another strategy, called the 2x2 strategy, which selects four test points for a single border segment. A pair of test points, one ON-point and one OFF-point, is chosen close to each vertex of the border segment. It has been proven that the BSE of the 2x2 strategy is always less than or equal to the BSE of the 2x1 strategy [2]. Moreover, the 2x2 strategy guarantees a finite BSE when the given border segment and the both given adjacent borders are closed.

2.4 An Alternative Method to Select Test Points.

ON-points of the White and Cohen 2x1 strategy should be in the domain satisfying all the path conditions. By relaxing some of those constraints we can get a better set of test points.

Suppose we have to select test points for a border segment of a given path domain. Let us consider the subpath

from the initial statement of the path up to the predicate which corresponds to the border segment to be tested. Then the path domain is a subset of that subpath domain and the border segment is a subset of the border segment of the subpath domain. For instance, consider a domain of a path with path predicates P_1, P_2, ..., P_5 according to their order in the execution of the path. The interpretations of the predicates are:

P_1 : $2y-x \leq 6$

P_2 : $y+3x \leq 27$

P_3 : $y \geq 1$

P_4 : $x \geq 2$

P_5 : $y - x \leq -2$

Two input variables x and y are constrained by the min-max condition $0 \leq x,y \leq 10$. The triangle FIJ in Figure 2.6 represents the corresponding domain. According to the min-max conditions on input variables, the input space is LMNO. To test the path domain, the border segments JF, FI and IJ which correspond to the predicates P_2, P_3 and P_5, respectively, should be tested. P_1 and P_4 are redundant predicates. The domain of the subpath up to the predicate P_2, is ACDO and the border segment JF of the given domain is a subset of the border segment CD of the subpath domain. Similarly the domain border segment IF is the subset of the border segment EF of ACEF which is the domain of the subpath up to the predicate P_3. Since the subpath domain corresponding to the predicate P_5 is the path domain itself,

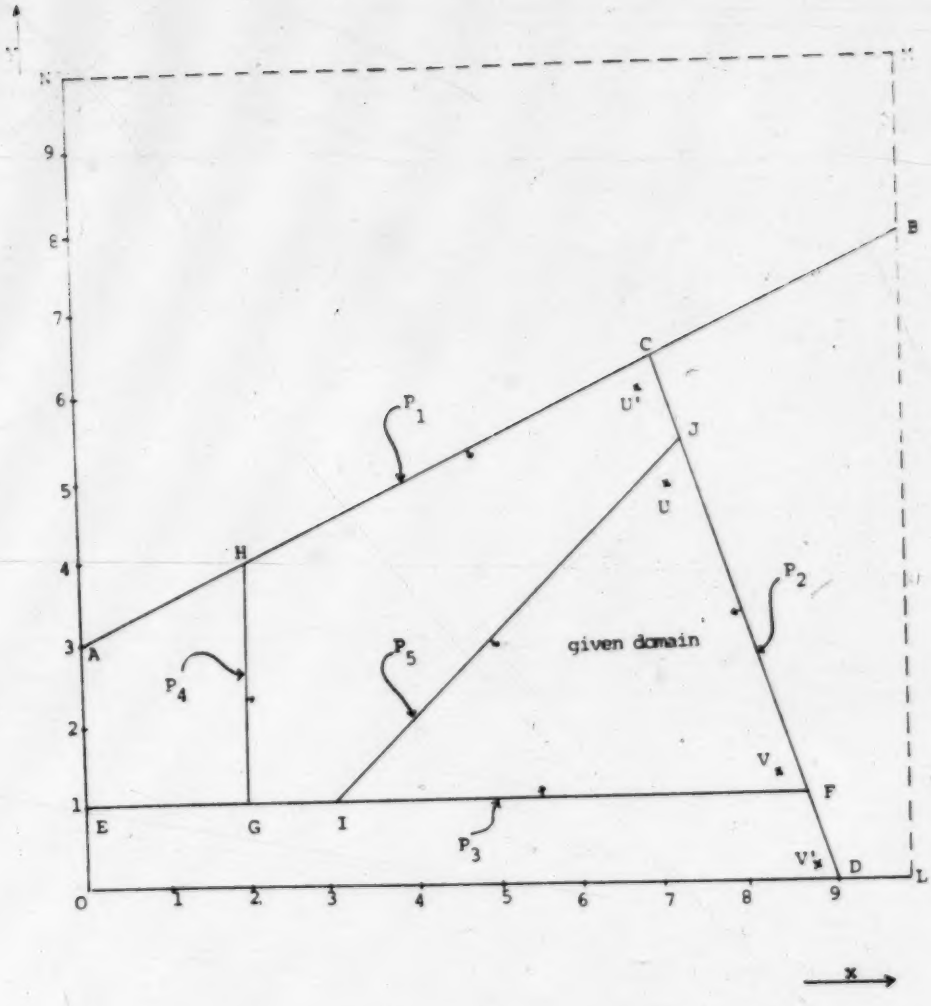


Figure 2.6 Subpath Domains

both border segments are the same line segment IJ.

Since the correctness of the domain border segment is implied by the correctness of the corresponding subpath border segment, we can test the border segment of the subpath domain instead of the given border segment. It generates more flexibility in selecting ON-points by relieving some constraints. For example, to test the border segment JF in Figure 2.6, we can select ON-points without considering the constraints that correspond to the predicates P_3, P_4 and P_5. Instead of U and V we can use U' and V' as the ON-points.

The advantage of the test points selected by this method can be stated formally by the following theorem:

Theorem:

Given a closed border segment of a path domain of a linearly domained program with a 2-dimensional input space, suppose the ON-points for 2x1 strategy are selected by following two different methods:

1. The ON-points should satisfy all the path conditions.
2. The ON-points should satisfy all the conditions of the subpath from the initial statement of the path up to the predicate which corresponds to the given border segment.

Assuming that the OFF-point would be the same and the ON-points are selected with an equal distance from the given

border, the BSE of method (2) is less than or equal to that of method (1).

Proof:

Since the path conditions for method (1) include all the path conditions for method (2), an ON-point selected by method (1) can be selected as an ON-point for method (2). Therefore, using method (2), we can select a set of two ON-points which is either the same set selected by the method (1) or a different set with a larger distance between the two points. If it is the same set of test points, the BSE will be the same. If the distance between the two points is increased, then the maximum angle between the given border and an ON-OFF line segment becomes smaller and the BSE may be decreased. This concludes the proof of the theorem.

CHAPTER 3

DOMAIN TESTING IN HIGHER DIMENSIONS

The domain testing strategy developed for programs with a 2-dimensional input space can be generalized to the N-dimensional case ($N \geq 3$). A path domain of a linearly domained program with an N-dimensional input space is a polyhedron which is the solution set of the conditions defined by the interpretations of the path predicates. Hence the dimension of a path domain is at most N. The boundary of an N-dimensional domain consists of (N-1)-dimensional hyperplane segments [5]. Each segment is a part of a hyperplane defined by the linear equation which corresponds to a predicate interpretation, and satisfies all the other path conditions. Such a segment of the domain boundary is called a *border segment* of the path domain. An (N-1)-dimensional hyperplane is uniquely determined by N linearly independent points on the hyperplane. Therefore, to identify the actual border segment, we need to find at least N linearly independent points on that border segment. Since any potential actual border must intersect every ON-OFF line segment we need at least N ON-OFF line segments. There are several techniques to achieve this, and three of the main strategies are discussed in this chapter.

3.1 The Nx1 Strategy.

The Nx1 strategy is a direct generalization of the 2x1 strategy to an N-dimensional input space. To obtain N ON-OFF line segments for a (N-1)-dimensional border segment, the Nx1 strategy requires one OFF-point and N linearly independent ON-points. Considering the linear independence of any set of N vertices of an (N-1)-dimensional convex polyhedron, White and Cohen [16] suggested the selection of ON-points at, or as close as possible to, the vertices of the border segment.

Unlike the 2-dimensional case, the number of vertices V of a border segment of an N-dimensional domain with $N \geq 3$ may not be equal to N. V strongly depends on the number of given adjacent border segments of the given border segment and is bounded by (see reference [8,9]):

$$B \leq V \leq \binom{B-[N/2]}{B-N+1} + \binom{B-[(N+1)/2]}{B-N+1} \quad -$$

where B is the number of given adjacent border segments which intersect the given border segment, forming (N-2)-dimensional hyperplanes and [x] is the greatest integer k such that $k \leq x$. Both terms of this upper bound of V are factorial numbers of B. Since $\binom{B-[N/2]}{B-N+1} \leq B^{N/2}$, the upper bound of V increases exponentially with N.

Thus a border segment of a domain with many path predicates can have a number of vertices V which is much greater than N, and there are $\binom{V}{N}$ possible sets of N vertices to select as the ON-points. Nevertheless, there is

no criterion developed to obtain a suitable set of vertices. Since any successful method should guarantee a good degree of reliability of the tested border, developing such a technique should be done on the basis of an error analysis.

3.2 Error Analysis in N-Dimensions.

3.2.1 BSE as the Error Measure in N-dimensions.

Analogous to the 2-dimensional case, the BSE in N-dimensions is the maximum N-dimensional volume of the input space which may be in a wrong domain due to a potential error of the border segment being tested. For a given set of Nx1 test data points, to determine the BSE, we have to consider the following N+2 positions of the actual border which limits the possible border shifts:

1. the hyperplane parallel to the given border through the OFF-point;
2. the hyperplane determined by the N ON-points;
3. the N hyperplanes determined by any set of N-1 ON-points and the OFF-point.

To determine the BSE we have to calculate the erroneous N-dimensional volume of the input space for each case. Suppose the predicate interpretations of the given border and the extreme actual border of the i'th case (i=1,2,3) are $\alpha^T X \leq a$ and $\beta_i^T X \leq b_i$ respectively. Here α and β_i are coefficient column vectors, X is the input column vector and a, b_1, b_2 and b_3 are scalar constants. Then the erroneous

volume for the first case is the volume of the polyhedron determined by $\alpha^T X > a$, $\beta_1^T X \leq b_1$ and the given adjacent borders. For the second case the required volume is the volume of the polyhedron determined by $\alpha^T X \leq a$, $\beta_2^T X > b_2$ and the given adjacent borders. Case (3) involves the volume of two polyhedra. An example in 3-dimensional space is given in Figure 3.1. The given border segment is the plane with vertices T,U,V and W. The planes KLTW, TUML, UVNM and WVNK are four given adjacent borders. The vertices U, V and W are selected as the ON-points. C' is the centroid of U, V and W. The OFF-point C is chosen at ϵ distance from C'. Then one of the limiting position of the actual border is PUQW, which is the plane determined by the ON-points U, W and the OFF-point C. Two tetrahedra UWQT and UWVP lie in wrong domains.

These two polyhedrons of case 3 are:

1. The polyhedron determined by $\alpha^T X > a$, $\beta_3^T X \leq b_3$ and the given adjacent borders.
2. The polyhedron determined by $\alpha^T X \leq a$, $\beta_3^T X > b_3$ and the given adjacent borders.

Therefore to determine the BSE for a given set of Nx1 test points we have to calculate the N-dimensional volume of 2N+2 polyhedrons. The N-dimensional volume $V_N(P)$ of a polyhedron P can be calculated by the recursive equation developed in [5]:

$$V_N(P) = \sum_{i=1}^{k} \{V_{N-1}(F_i) * d(x,F_i)\}$$

where x is any point in the polyhedron, F_i is a facet of P, $d(x,F_i)$ is the distance from x to F_i and k is the number of

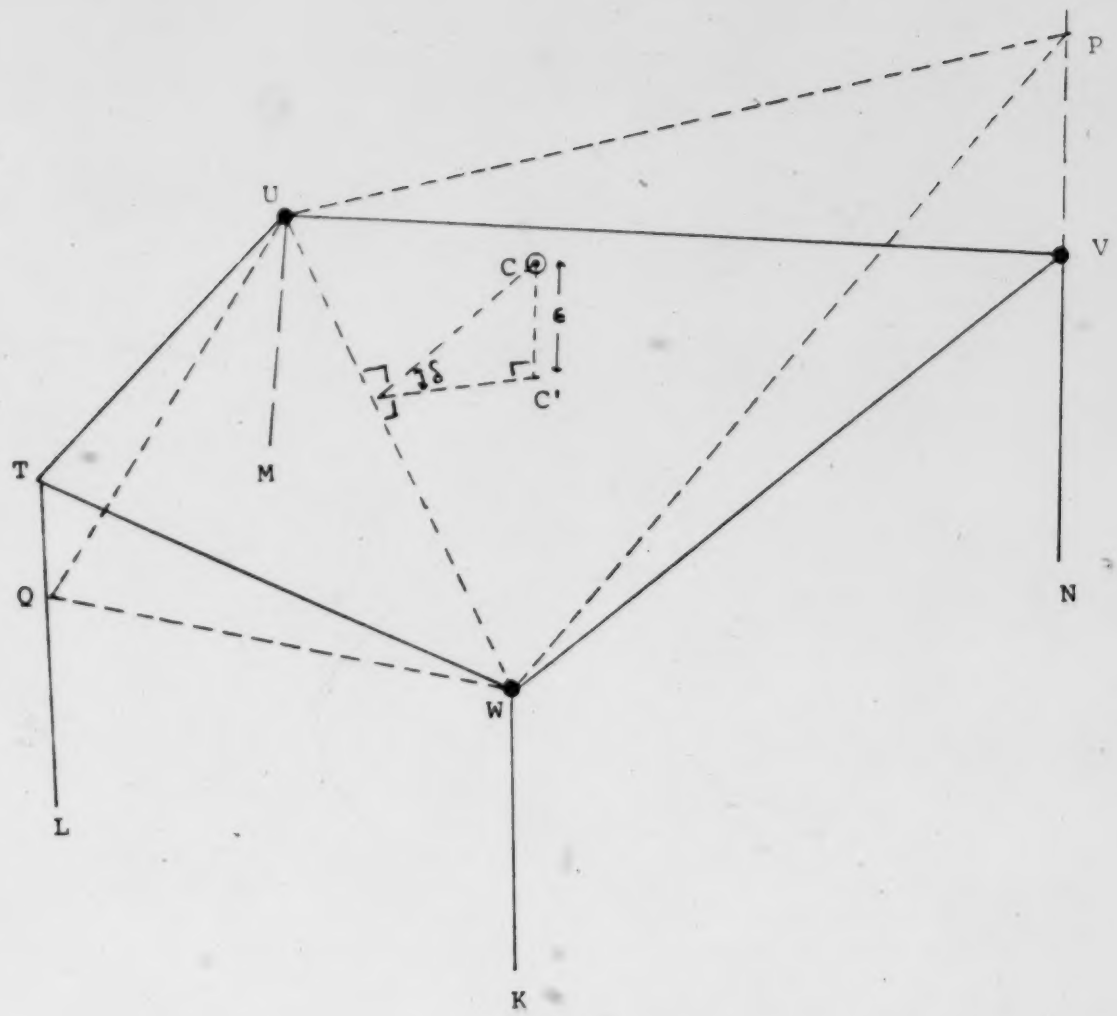


Figure 3.1 Undetectable Potential Border Shifts

facets of P. Hence calculating the volume of one N-dimensional polyhedron requires the i-dimensional volume of all the i-faces of the polyhedron for i=1,...,N-1. Since the BSE in the Nx1 strategy involves 2N+2 polyhedrons, calculating the BSE in higher dimensions is intractable.

3.2.2 An Alternative Error Measure in N-dimensions.

Due to the intractability of calculating the BSE for a given set of test data, it is necessary to define another criterion to represent the correctness of a test. In domain testing it is assumed that the the given border and a possible actual border are both linear. If the input variables are X_1, X_2, ... X_N, suppose the predicate interpretation of the given border is

$$a_1X_1 + a_2X_2 + \ldots\ldots + a_NX_N = a_0$$

and the equation of the actual border is

$$b_1X_1 + b_2X_2 + \ldots\ldots + b_NX_N = b_0$$

where a_i and b_i (i=0,1,2,...,N) are constants. If $a_i=k*b_i$ for all i = 0,1,2,.....,N for some constant k then the given border is the actual border. If it is incorrect there can be two types of error.

1. If $a_i=k*b_i$ for i=1,2,....,N and $a_0 \neq k*b_0$ the given border is shifted parallel to the actual border. We call this error a *Parallel Displacement* of the tested border.
2. If there exists no constant k such that $a_i=k*b_i$ for all i in {1,2,....,N} the given border is

inclined, making an angle with the actual border. We call this error an *Angular Displacement* of the tested border.

For a set of Nx1 test data, parallel displacement towards the outside of the domain is constrained by the distance from the OFF-point to the given border. The parallel displacement towards the inside of the domain is constrained by the minimum distance from an ON-point to the given border. The maximum of these two limits is called the *Maximum Parallel Displacement (MPD)* of the tested border. The angular displacement cannot exceed the maximum angle between the given border and a limiting position of the possible actual border. This limit is called the *Maximum Angular Displacement (MAD)* of the tested border.

Since an error in any element of the coefficient vector can cause an angular displacement of the border segment, we can expect a higher probability of an angular displacement than a parallel displacement. Therefore the MAD is more appropriate as an error criterion than the MPD.

Unlike the BSE, the calculation of the MAD is not difficult in higher dimensions. The angle between two hyperplanes is given by the angle between the normal vectors of the hyperplanes. As shown in section 3.2.1, there are N limiting positions of the possible actual border which cause an angular displacement for the Nx1 strategy. These are the hyperplanes determined by any set of N-1 ON-points and the OFF-point. The MAD of the tested border is the maximum angle

between the normal vector of an extreme hyperplane and the normal vector of the given border. Since we know N points on each hyperplane, we can find these normal vectors by solving N linear simultaneous equations. Thus the MAD of the selected Nx1 test points can be calculated by solving N systems of N linear equations.

Although it can be determined in a polynomial time for a set of given Nx1 test data, the MAD does not reflect how much of the input space can be processed incorrectly due to an error of the tested border. If we use the MAD as the error measure, we assume that the potential error is independent of the external angles of the given adjacent borders. Moreover,like the BSE, the MAD is insensitive to redundant predicates and the boundary of the input space.

There is another aspect of the error criterion to be considered when determining the acceptability of a test. An unbounded volume of input space may still be a negligible fraction of the entire program input space. Therefore, an unbounded BSE may not indicate that the test is insignificant. Figure 3.2 illustrates the situation in 2-dimensions. The unbounded area is given by the area between lines L_1 and L_2. The angle between L_1 and L_2 in Figure 3.2(a) is small and the area is a small fraction of the whole space. In Figure 3.2(b) the angle between two lines is large and the area is a significant fraction. The significance of the error is represented, not by the area, but by the angle. Therefore, it may be more reasonable to

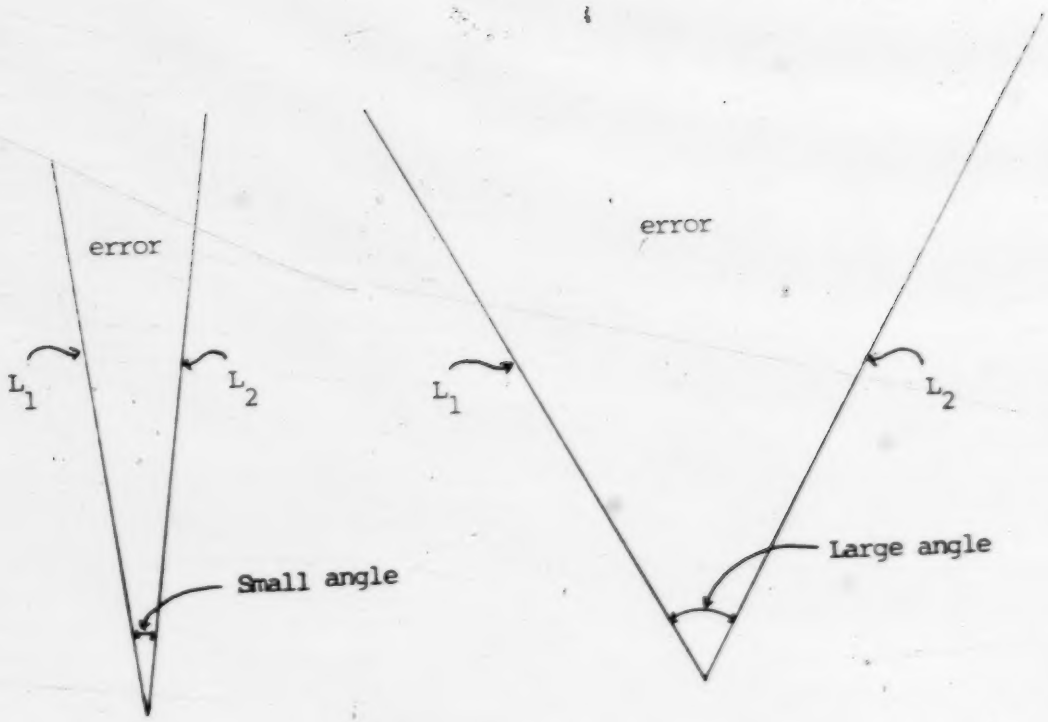


Figure 3.2 Two Possible Situations of Unbounded Error

reject a domain test on the basis of a large MAD than the unbounded BSE.

The MAD of the tested border can be used as the error criterion to analyze domain testing methods in higher dimensions. Then we have to select test points to minimize the MAD of the tested border. Since the BSE depends on the angles between the given border and the extreme positions of the actual borders, we can expect a small BSE for a set of test data which minimizes the MAD of the tested border.

To study the possible methods of selecting test points to reduce the MAD, we consider a geometrical interpretation of the MAD. H_1 and H_2 in Figure 3.3 are two hyperplanes in a 3-dimensional space. They intersect in the line AB. C is a point on H_1 and C' is the projection of C on H_2. The distance from C'to AB is r and $CC'=\epsilon$. Then the angle δ between H_1 and H_2 is given by $\delta = \tan^{-1}(\epsilon/r)$. As shown in Appendix A it is valid in any N-dimensional space with $N\geq 2$.

We can use this concept to minimize the MAD for the Nx1 strategy. First we consider a border segment in a 3-dimensional input space. As shown in Figure 3.4, to test a border segment ABCDE of a 3-dimensional domain, the three vertices A, B and C are selected as the ON-points. W is any point in the triangle ABC on the border segment. The OFF-point is selected at ϵ distance from W. Then the MAD is the maximum angle between the given border segment and a plane determined by the OFF point and any two ON-points. Using the above geometrical interpretation it can be

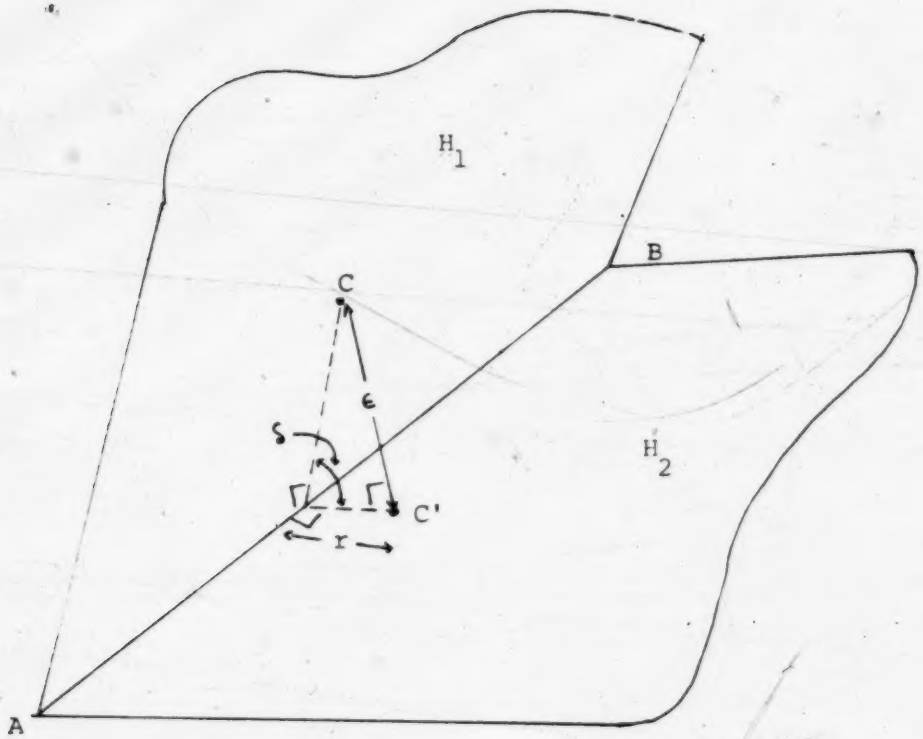


Figure 3.3 A Geometrical Interpretation of the MAD

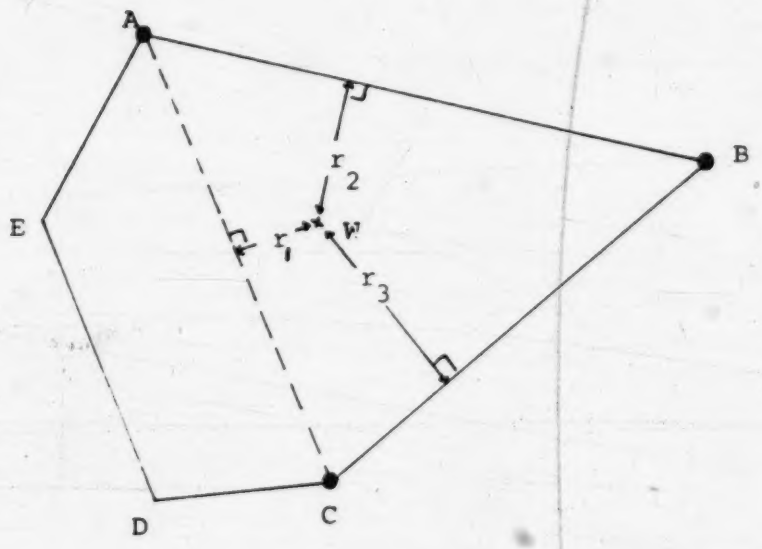


Figure 3.4 Optimizing the MAD for the 3x1 Strategy

formulated as:

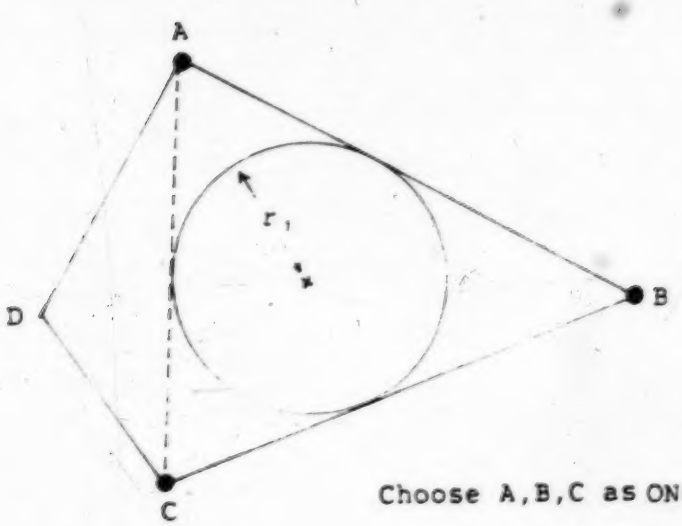
MAD = Maximum of $\{\tan^{-1}(\epsilon/r_1), \tan^{-1}(\epsilon/r_2), \tan^{-1}(\epsilon/r_3)\}$

Therefore, the MAD can be reduced by two methods. One method is to reduce ϵ. This can be done by selecting the OFF-point as close to the given border as possible.

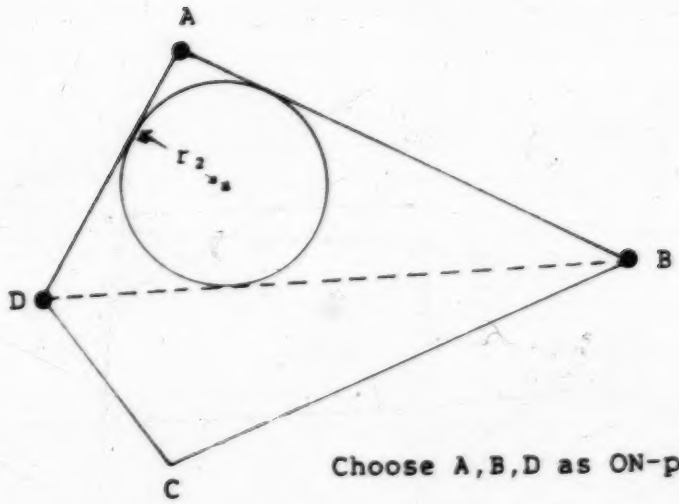
The other method is to reduce the maximum of $\{1/r_1, 1/r_2, 1/r_3\}$. In other words we should increase r which is the minimum of $\{r_1, r_2, r_3\}$. For the selected ON-points A,B and C, r can be maximized by selecting W to be the *incenter*, the center of the inscribed circle of the triangle ABC.

Therefore, to optimize the MAD of the 3x1 strategy the ON-points should be the set of three vertices which contains the largest circle in the triangle formed by them. Also, the OFF-point must be selected as close to the incenter of that circle as possible.

In general, the MAD of the Nx1 strategy is optimized when the *simplex*, the (N-1)-dimensional polyhedron with N vertices, formed by N ON-points contains the largest (N-1)-dimensional sphere and the OFF-point is close to the incenter of the simplex. A set of ON-points selected in this manner is called a more *scattered set*. As shown in Figure 3.5, on a 2-dimensional border of a 3-dimensional domain, the three ON-points should be selected such that the triangle formed by them contains the largest inscribed circle. Since $r_1 > r_2$, {A,B,C} gives a smaller MAD than {A,B,D}. Since finding the incenter of a simplex is difficult we can use the centroid to select the OFF-point.



Choose A,B,C as ON-points



Choose A,B,D as ON-points

Figure 3.5 Selecting Vertices as ON-Points

The centroid is the average of the vertices of a simplex and is close to the incenter; both are the same for an equilateral simplex.

3.2.3 A Sufficient Condition for Finite Error in Higher Dimensions.

In a 2-dimensional space, to determine whether the BSE is finite we have to consider the intersection of the possible actual borders with the extended given adjacent borders. This can be done by comparing the external angles of the given adjacent borders with the angles between the ON-OFF lines and the given border. Nevertheless, in a higher dimensional space the intersection of the actual borders and extended given adjacent borders does not guarantee that the BSE is finite.

An N-dimensional polyhedral domain has i-dimensional faces (i-face) for i=0,1,2,...,N-1. A portion of the domain border which corresponds to k predicates for some k=1,2,...,N is an (N-k)-face. Thus a vertex is a 0-face and an edge is a 1-face. A border segment is an (N-1)-face. For a given border segment, it is possible that an adjacent k-face for some k<(N-1) may not intersect a possible actual border causing an unbounded BSE while all the given adjacent border segments intersect all the possible actual borders.

As an example, consider the tetrahedral domain shown in Figure 3.6. Suppose H, H_1, H_2 and H_3 are the planes containing triangles ABC, BCD, ABD and ACD respectively. To

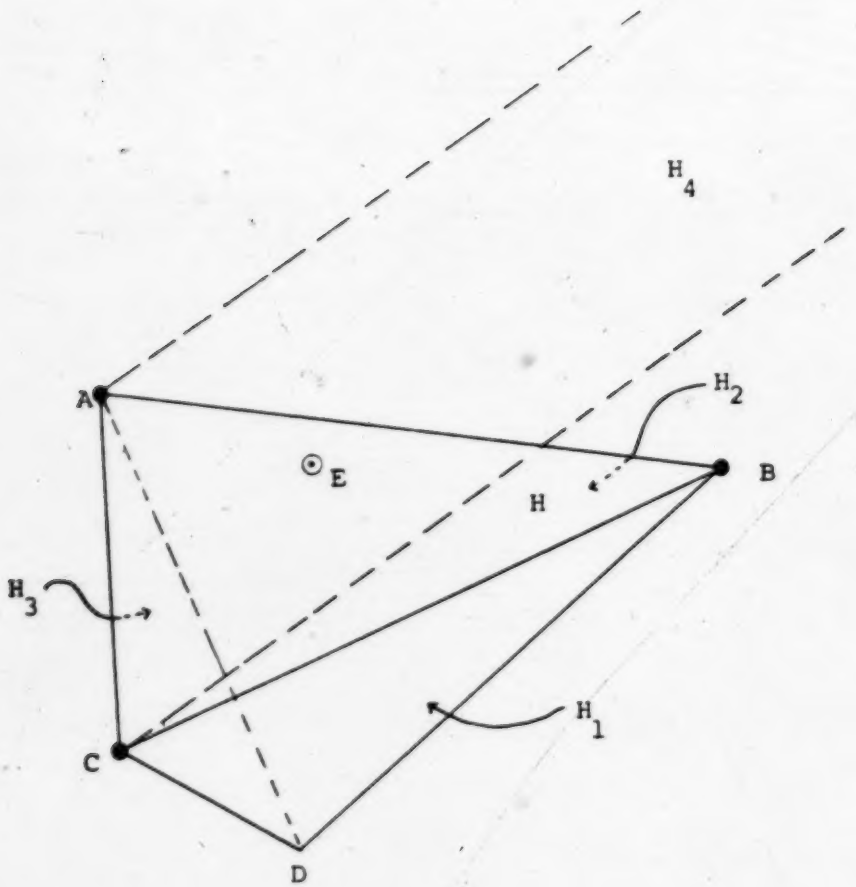


Figure 3.6 Nearly Parallel Adjacent Edges

test H the three vertices A, B and C are selected as the ON-points. The OFF-point E is selected at a small distance from the centroid of A,B and C. Suppose the plane through A, C and the OFF-point E is H_4 and it is parallel to the edge DB. Then H_4 intersects the given adjacent borders H_1, H_2 and H_3 but does not intersect the edge DB, causing the space between H, H_1, H_2 and H_4 to be unbounded. That is, the BSE is infinite.

The border of an i-face for some i=1,2,...,N-1 is formed by a set of (i-1)-faces. Therefore, if a actual border is intersected by all the extended (i-1)-faces, then we can conclude that the actual border is intersected by all the i-faces. i.e., the intersection of an actual border by all the higher dimensional faces is determined by the intersection of that border by all the edges (1-dimensional faces).

Thus in an N-dimensional input space unbounded BSE is caused by nearly parallel adjacent edges. Therefore to determine whether the BSE is finite we have to examine the external angles of all the edges adjacent to the given border segment. Since we know the unit normal vector of the given border segment, we can determine the external angle of an adjacent edge if we know the direction of the edge. An edge of an N-dimensional polyhedron is the intersection of a set of N-1 hyperplanes. Since the edge is an straight line on all these hyperplanes, it is perpendicular to the corresponding unit normal vectors. Using this fact we can

obtain a set of N-1 linear equations whose solution is a straight line parallel to the edge.

For instance consider the edge DB of the 3-dimensional polyhedron ABCD in Figure 3.6. It is the intersection of the two planes H_1 and H_2. Therefore, DB is perpendicular to the unit normal vectors of H_1 and H_2. Solving the two equations obtained by these relationships we can find the direction of DB.

Since there should be at least one edge through each vertex, a border segment of an N-dimensional domain has at least V edges. Hence, to test for finite BSE we have to determine the external angles of at least V edges which can be a very large number in higher dimensions. Although it is expensive, we can test for a finite BSE by considering all the adjacent edges by this method.

Using this concept we can obtain a sufficient condition for finite BSE in terms of the MAD. The MAD of a set of test points gives the largest angle between the given border and a potential actual border. If the MAD is less than the smallest angle between the given border and an adjacent edge then all the extended adjacent edges will be intersected by all potential actual borders. Therefore, a sufficient condition for finite BSE in higher dimensions is

$$\text{MAD} < \theta$$

where θ is the smallest angle between the given border segment and an adjacent edge.

3.3 Other Strategies.

3.3.1 The NxN Strategy.

We can improve the accuracy of the Nx1 strategy with respect to either the BSE or the MAD by selecting N OFF-points instead of one. This approach is called the NxN strategy. For the NxN strategy an OFF-point is selected at ϵ distance from each ON-point. A limiting case of the possible actual border in the NxN strategy is a hyperplane determined by a set of N-1 ON-points and the OFF-point close to the remaining ON-point. The tangent of the angular displacement of this hyperplane is the ratio of the distance from the OFF-point to the given border and the distance from the remaining ON-point to the (N-2)-dimensional manifold through the N-1 ON-points. The latter distance is longer than the distance to the centroid from the N-1 On points in the Nx1 strategy. Therefore, the MAD of the NxN strategy is always less than the MAD of the Nx1 strategy.

The example in Figure 3.7 illustrates this possibility in 3-dimensions. Suppose three ON-points A, B and C are selected an equal distance from each other. The OFF-point for the Nx1 strategy is selected at an ϵ distance from the centroid D of A, B and C. Then for the Nx1 strategy, the MAD δ_1 is $\tan^{-1}(\epsilon/d)$. If we select three OFF-points at A, B and C with the same distance ϵ, the MAD δ_2 is $\tan^{-1}(\epsilon/3d)$. For small angles this relation implies $\delta_2 \cong \delta_1/3$. It is a significant improvement of the MAD caused by two additional

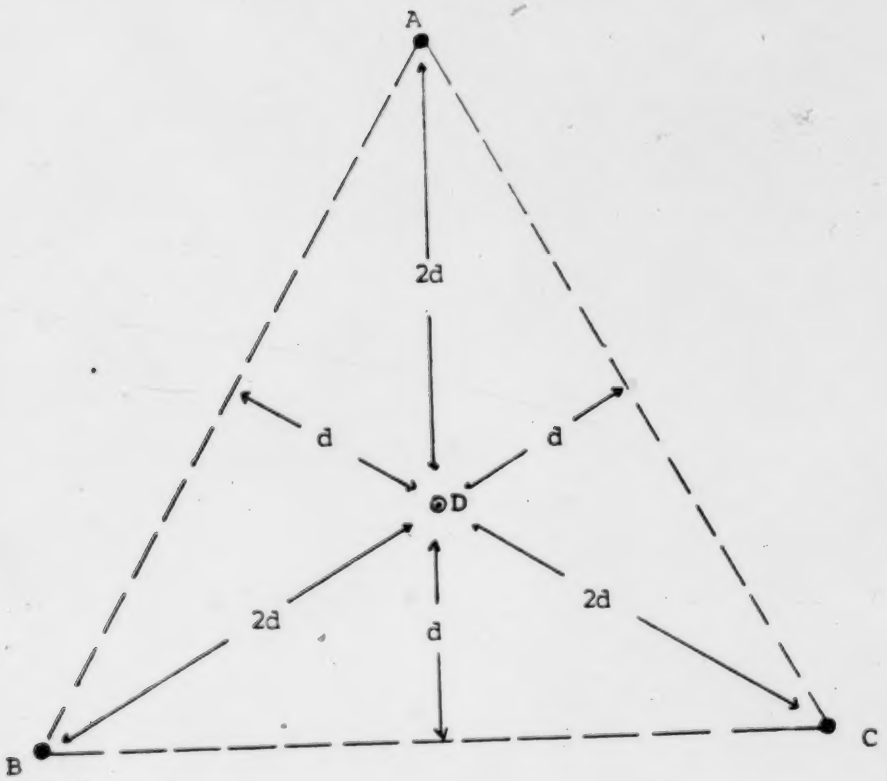


Figure 3.7 Calculating the MAD in 3-Dimensions

test points.

3.3.2 The VxV Strategy.

The Clarke et al. VxV strategy [2] recommended the selection of an ON-point at each vertex of the border segment and an OFF-point at each vertex of the hyperplane segment parallel to the given border at an ϵ distance. The objective of the VxV strategy is to guarantee a finite BSE. We can expect that the VxV strategy guarantees a finite BSE if we can select all the vertices of both hyperplane segments as ON and OFF-points. Then there is a pair of ON and OFF-points on each edge through all the vertices of the given border segment. Therefore, any hyperplane separating the ON-points and the OFF-points will intersect each extended adjacent edge and the BSE is finite. If there is at least one given adjacent border which is open, then we cannot select a pair of ON and OFF-points on the edges that lie on the open given adjacent border. Then there can be a potential actual border which does not intersect such an edge. Thus a finite BSE for the VxV strategy cannot be guaranteed.

The set of ON-points of the NxN strategy is a subset of the ON-points of the VxV strategy and the set of OFF-points of the NxN strategy is a subset of the OFF-points of the VxV strategy. Thus the VxV strategy uses all the test points used for the NxN strategy and several additional test points. Therefore either the BSE or the MAD of the VxV

strategy should be less than those of the NxN strategy. Since there is a large number, possibly $\binom{V-1}{N-1}$, of extreme positions of the possible actual border, it is difficult to calculate the MAD for the VxV strategy.

3.4 Evaluation of the Strategies.

A straightforward method to evaluate a testing method is to consider the number of test points required for a test. Suppose we want to test an N-dimensional domain whose border consists of B (N-1)-dimensional hyperplanes. Then B is less than the number of path conditions of the domain which is independent of N. To test a border segment by the Nx1 strategy, N+1 test points are required and the whole domain can be tested by at most B*(N+1) test points. The NxN strategy requires at most 2*B*N test points. If V_i is the number of vertices of the i th border segment, then the VxV strategy requires at most $2*\Sigma V_i$ test points. Thus the Nx1 and the NxN strategies require a number of test points on the order of N. The number of test points for the VxV strategy is nonpolynomial in N and B, since it depends on the number of vertices which may be a factorial number of N and B for N > 2.

There are two major components to be considered when evaluating the time complexity of a test for a selected path:

1. the time required to find the set of test data points;

2. the time required to execute the program for each test point.

Given a domain to be tested we know all the path predicates, hence we know the equations of the (N-1)-dimensional hyperplanes which determine the domain. We do not have any knowledge about the vertices of the domain. To test a border segment the VxV strategy requires the coordinates of all the vertices of the border segment. Since there should be at least N hyperplanes through a vertex of an N-dimensional polyhedron, a vertex of the given border segment is the intersecting point of at least N-1 given adjacent borders and the given border. Therefore, the intersecting point (if one exists) of any N-1 hyperplanes and the given border is a candidate for a vertex. There are $\binom{B-1}{N-1}$ cases to be considered. Although some of these points may not be vertices of the given border segment, we cannot identify them without applying the path conditions for each point. In Figure 3.8 ABCDE is a border segment with five given adjacent borders in 3-dimensions. Since the points F, G, H, I and J do not satisfy all the path conditions, they are not the vertices of the given border segment. Therefore, to find all the vertices we have to solve $\binom{B-1}{N-1}$ sets of N simultaneous equations and apply all the path conditions to each solution. The time required to find all the vertices is greater than the time required to solve $\binom{B-1}{N-1}$ sets of N equations. Since solving a set of N linear equations can be done in polynomial time on the order of N^3, the order of the

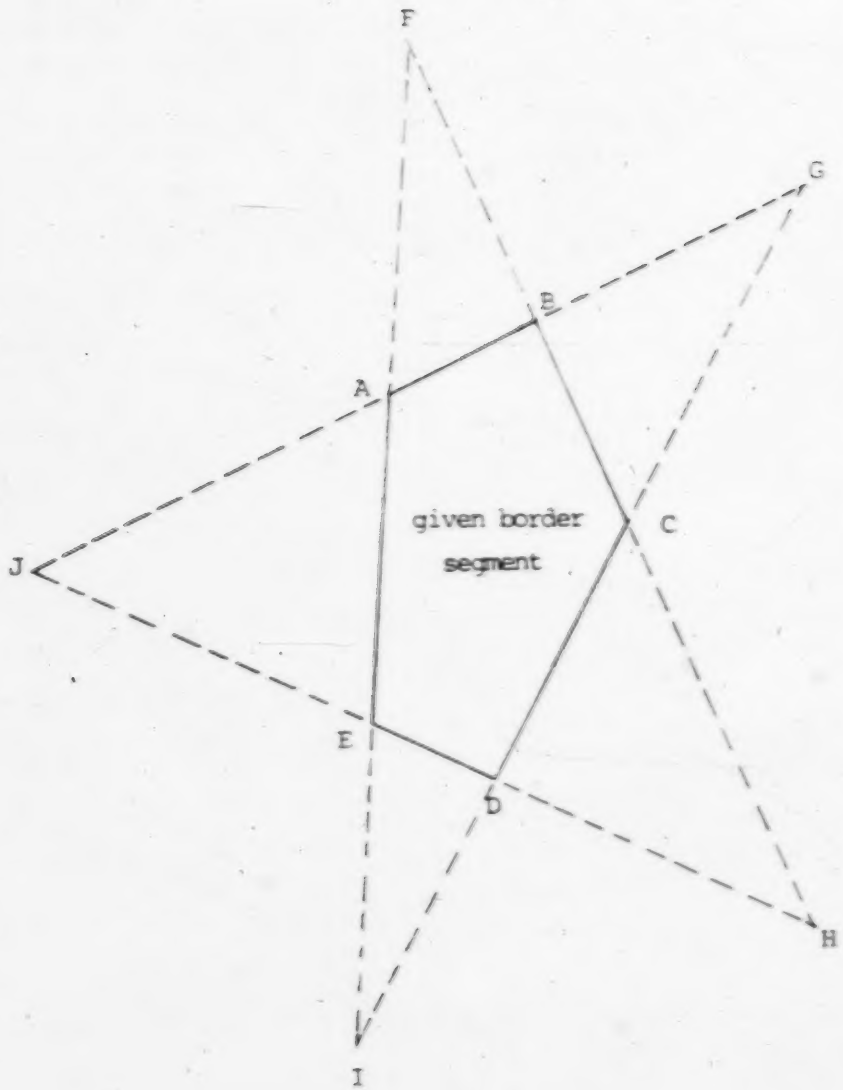


Figure 3.8 Vertices not Adjacent to the Border Segment

amount of time required to find all the vertices is as high as $N^3*B!$.

The time required to execute the program for all the vertices in the VxV strategy is 2*V*T where T is the time required to execute the program once. Since V is O(B!) for higher dimensions, the complexity of this time is O(B!). Therefore, the time complexity of the VxV strategy is $O(N^3*B!)$ for the worst case.

To test a whole domain, the Nx1 strategy requires at most B*(N+1) points to be tested and the NxN strategy requires 2*B*N points. Therefore, the second portion of the time for both strategies is O(N*B). To choose test data both methods require the selection of N vertices. If we want to find the set of N vertices which gives the smallest MAD, it cannot be done without finding all the vertices. The time complexity of the best Nx1 or NxN strategy becomes $O(N^3*B!)$ and it is the same as the VxV strategy from which we can expect more accuracy.

In domain testing it is assumed that an "oracle" exists which can always determine whether a specific test case has been computed correctly or not. In reality, the tester himself must make this determination and the time spent examining and analysing the output of test data is a major cost. Hence, the order of the number of required test points is an important factor which should be considered when a test method is developed.

3.5 An Alternate Strategy.

In considering the possibility of developing a testing method with polynomial time complexity, the VxV strategy can be of no assistance. It involves the number of test points 2*V which is a factorial number of B. The Nx1 and the NxN strategies can be applied in polynomial time if test points can be selected in polynomial time. The time complexity of selecting N vertices is non-polynomial, if we select the set of N vertices with the smallest MAD. Since a more scattered set of ON-points gives a smaller MAD, it is a good heuristic to select a set of test data whose MAD is close to the best. For this purpose the following algorithm has been developed.

Suppose a border segment determined by the predicate $a^T X \geq a$ should be tested. To test the border segment the given predicate should be non-redundant. The redundancy of a predicate can be verified by minimizing the objective function $a^T X - a$ subject to the path conditions. For a non-redundant predicate the minimum of $a^T X - a$ should be zero. If it is a non-redundant predicate, a set of ON-points can be chosen as follows:

1. Select any objective function $\beta_1^T X$ such that $\beta_1^T X = 0$ is perpendicular to $a^T X = a$. Minimizing and maximizing $\beta_1^T X$ subject to the path condition, we can find two points V_1 and V_2.
2. After the (k-1)'th point $V_{(k-1)}$, (2<k<N), is selected, define the objective function $\beta_k^T X$

satisfying the following two conditions:

a. $\beta_k^T X=0$ is parallel to the (k-2)-dimensional manifold determined by $V_1, V_2, \ldots, V_{k-1}$.

b. $\beta_k^T X=0$ is perpendicular to $a^T X=a$.

Then $\beta_k^T X$ is minimized and maximized subject to the path conditions. It should produce another ON-point V_k or two other ON-points V_k and V_{k-1}.

3. Repeat step 2 until the set of ON-points contains N or more points.

Since $\beta_1^T X=0$ is perpendicular to $a^T X=a$, we can obtain β_1 by finding a solution for $a^T \beta_1=0$. To select β_k for the second step of the algorithm, we have to find a solution for the following set of k linear equations:

$$\beta_k^T V_i = \text{constant for all } i=1,2,\ldots,k-1$$

$$a^T \beta_k = 0.$$

Since we have a set of k linear equations in N-dimensions we can obtain a solution by giving arbitrary values for (N-k) elements of β_k.

We should note that a point selected by this algorithm may not be a vertex of the border segment. The objective of selecting vertices as ON-points is to obtain a set of N linearly independent ON-points. As described in the second step of the algorithm, to select the k'th point V_k we optimize an objective function which is parallel to the (k-2)-dimensional manifold defined by $V_1, V_2, \ldots, V_{k-1}$. Therefore, V_k cannot be a point on that manifold and $V_1, V_2,$

....., V_k are linearly independent. Thus, the algorithm guarantees a set of N linearly independent ON-points which are scattered over the given border segment.

This algorithm produces N or N+1 points and we call it the *Scattering Method*. A set of ON-points can be selected using this algorithm by applying a linear programming technique at most N times. Since linear programming can be done in polynomial time, the complexity of selecting the set of vertices is polynomial in N and B. Another advantage of this strategy is that it does not need any prior knowledge of vertices. The only requirement is the predicate interpretations of the path predicates.

A better accuracy can be obtained by selecting N or N+1 OFF-points instead of one. Since the OFF-points are selected at ϵ distance from the given border, they should be selected on the hyperplane segment which is determined by $\alpha^T X=a-\|\alpha\|\epsilon$. Also they should satisfy all the path conditions except $\alpha^T X\geq a$ which is the predicate related to the given border segment. Thus to select the OFF points, the same algorithm can be used.

As an example of this method in 3-dimensions, consider the border segment ABCDEF of Figure 3.9. For the first step suppose we select the objective function $\beta_1^T X$ where $\beta_1^T X=0$ is parallel to an arbitrary line GH on the given border and perpendicular to the hyperplane. Maximizing $\beta_1^T X$ we get the vertex A and minimizing $\beta_1^T X$ we get the vertex D. Then the next objective function $\beta_2^T X$ should be parallel to AD and

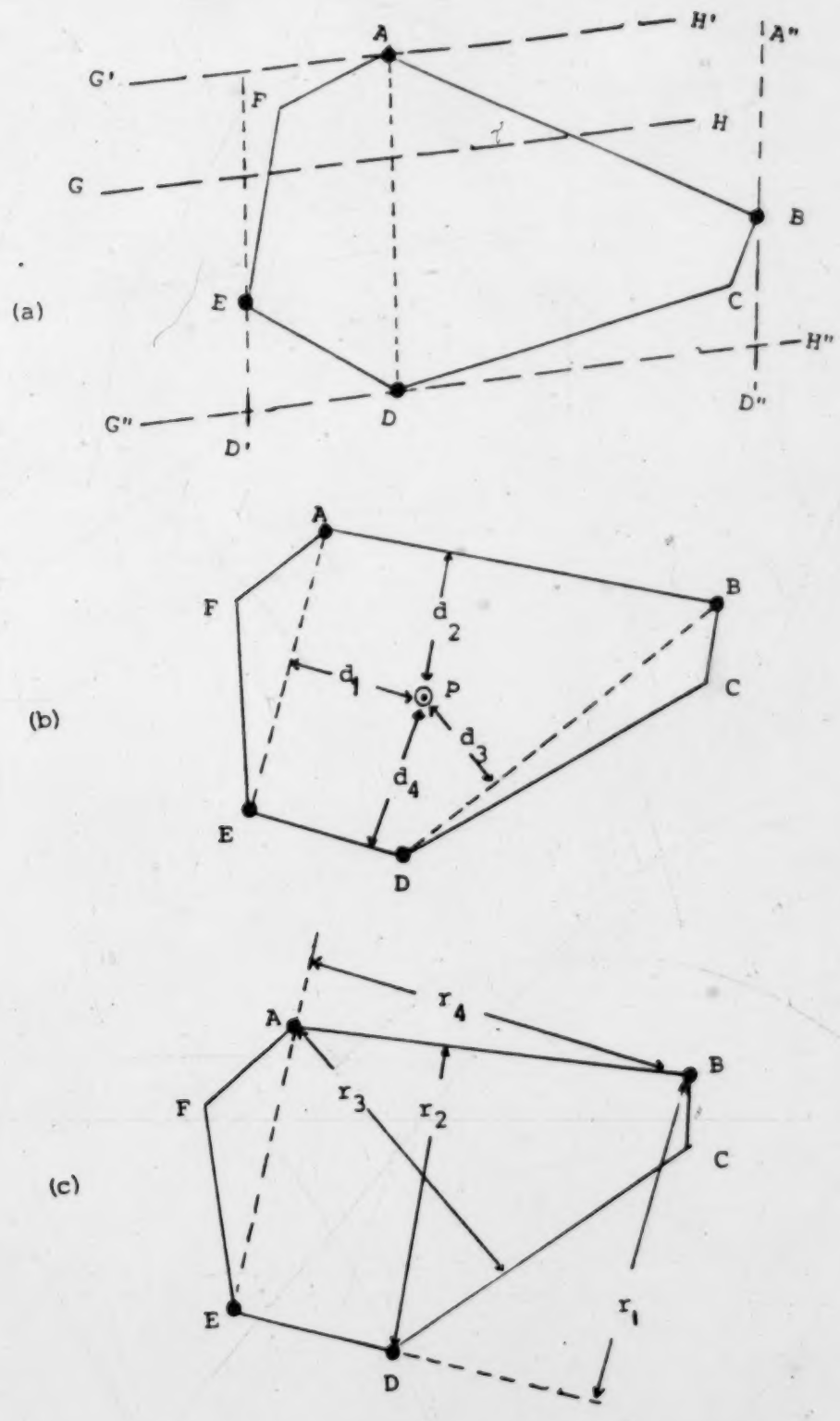


Figure 3.9 Scattering Method in 3-Dimensions

perpendicular to the border segment. When we maximize and minimize $\beta_2^T X$ subject to the path predicates we will find the vertices B and E. Therefore, we may select {A,B,D,E} as the set of ON points to test the border segment. In Figure 3.9(b) P is the centroid of {A,B,D,E}. If we select the OFF-point at ϵ distance from P, the MAD of the tested border is the maximum of $\{\epsilon/d_1, \epsilon/d_2, \epsilon/d_3, \epsilon/d_4\}$. If we select an OFF-point close to each ON-point (Figure 3.9(c)), then the MAD of the tested border is the maximum of $\{\epsilon/r_1, \epsilon/r_2, \epsilon/r_3, \epsilon/r_4\}$.

To obtain a better accuracy, we can generalize the method developed in section 2.4 to select ON-points in N-dimensions. Then the Scattering Method should be applied with the corresponding subpath conditions instead of all the path conditions.

3.6 An Upper Bound of the MAD of the Scattering Method in 3-Dimensions

The algorithm, when applied in 3-dimensions, generates a parallelogram which contains the given border segment. Figure 3.10 shows the parallelogram generated for the above example. PQ and RS are intersecting lines of the given hyperplane with $\text{Max}\{\beta_1^T X\}$ and $\text{Min}\{\beta_1^T X\}$ respectively. Similarly, QR and PS are determined by the objective function $\beta_2^T X$.

Each side of the parallelogram should contain at least one point selected by the algorithm. Hence, we can use the

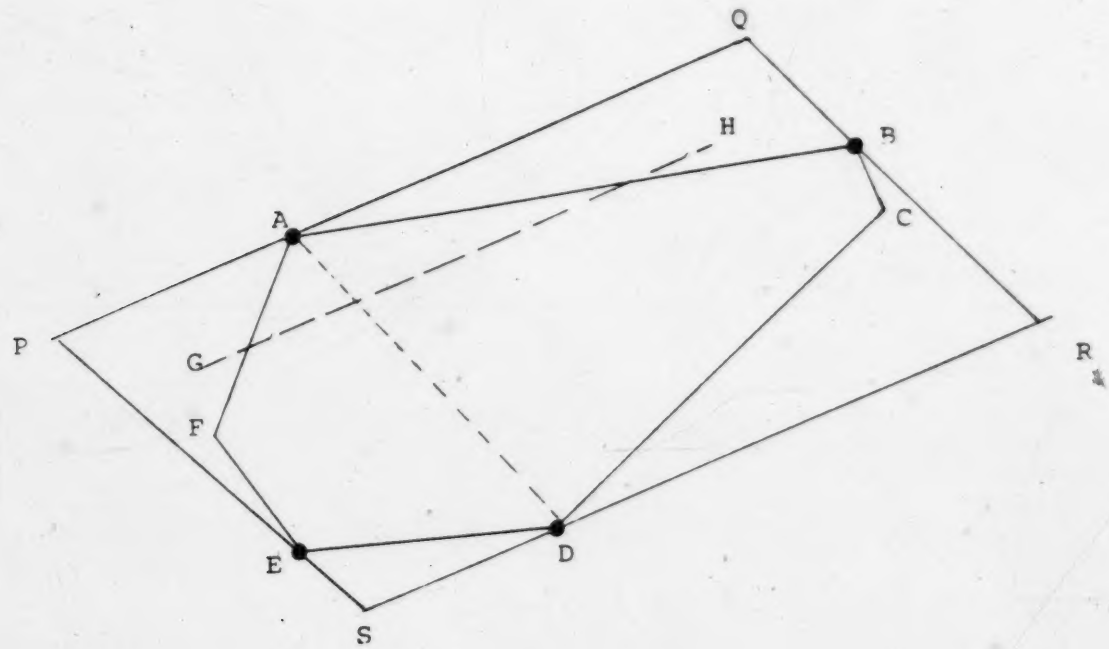


Figure 3.10 Parallelogram Generated by the Scattering Method in 3-Dimensions

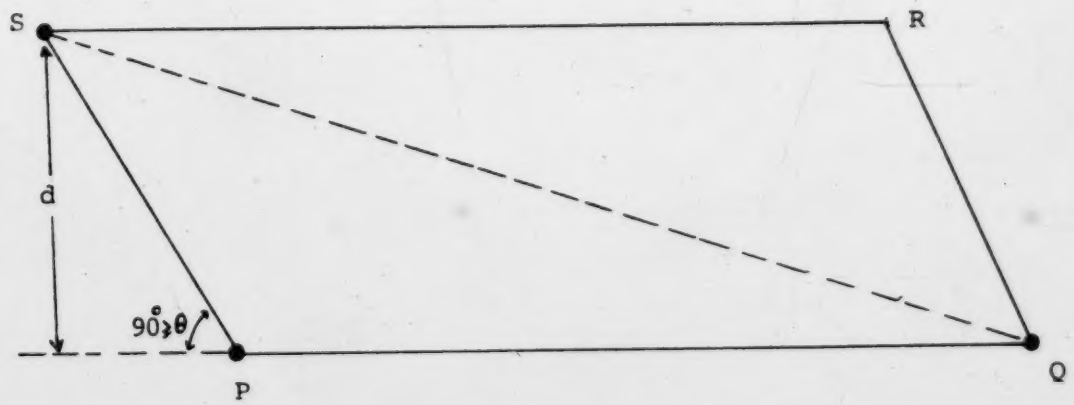


Figure 3.11 Worst Case for ON-points

dimensions of the parallelogram to obtain an upper bound for the MAD of the test. Since the MAD depends on the maximum radius of the circle contained in the triangle (if three vertices were found), or the quadrilateral (if four vertices were found), whose vertices are the selected points, we can obtain an upper bound for the MAD in terms of the dimensions of the parallelogram. The worst case is depicted in Figure 3.11. The parallelogram generated by the algorithm is PQRS and the selected vertices are P, Q and S. The angle QPS $\geq 90°$. d is the smallest distance between two parallel lines. In the Appendix B it is proven that the MAD is less than or equal to $\tan^{-1}(6\epsilon/d)$ if the OFF-point is selected at ϵ distance from the centroid of PQS. Since this is the worst case, the algorithm guarantees that the MAD in 3-dimensions is less than $\tan^{-1}(6\epsilon/d)$.

3.7 Summary

This chapter has surveyed the application of domain testing in an N-dimensional input space when N>2. A path domain of a linearly domained program with N input variables is a polyhedron whose dimension is at most N. We need N ON-OFF line segments to test a border segment of an N-dimensional path domain since it is an (N-1)-dimensional hyperplane segment. The number of vertices V of such a border segment is greater than the number of given adjacent border segments B and bounded above by a factorial number of

B and N.

To determine the BSE of a test in higher dimensions is very complex. It involves calculating the N-dimensional volumes of 2N+2 polyhedrons. We have discussed the possibilities of an unbounded BSE of a test. Although it is expensive, we can determine whether the BSE is finite or not by considering all the adjacent edges of the border segment.

Due to the complexity of the BSE as the error measure in higher dimensions, we have developed a new error criterion called the Maximum Angular Displacement (MAD). The MAD of the tested border can be determined in polynomial time. It has several drawbacks as an error measure in domain testing. It is independent of the given adjacent borders of the given border segment. Also, it does not give an upper bound for the potential error as the BSE does. Nevertheless, the MAD is not difficult to determine in higher dimensions and we can expect a small BSE for a test with a small MAD.

We examined three major domain testing methods, the Nx1, NxN and VxV strategies proposed by [2,15]. Although the Nx1 and NxN strategies need a set of N vertices as ON-points, neither of them provides sufficient guidance in choosing N out of V vertices. If the same set of ON-points is selected for both methods then the accuracy of the NxN strategy is better than that of the Nx1 strategy. Considering the accuracy, the VxV strategy cannot be worse than NxN method, but it is difficult to see a significant improvement.

These methods have been evaluated on the basis of the number of required test points and the time complexity in terms of the dimension N and the number of path predicates B. The number of test points required for the VxV strategy is non-polynomial while that of the Nx1 and NxN strategies is polynomial. All three strategies have non-polynomial time complexities.

The major achievement of this chapter is to develop the Scattering Method to select test data. It does not provide the best test points but a good set with a small error. At the cost of an additional N or N-1 test points, we can obtain a better accuracy by selecting an OFF-point for each ON-point.

This chapter concludes with an upper bound obtained for the MAD of the Scattering Method in 3-dimensions.

CHAPTER 4

DOMAIN TESTING IN DISCRETE INPUT SPACE

4.1 Discrete Input Space.

Since a finite number of binary bits cannot be used to represent all the numbers on a continuous line segment, in reality the input space of a computer program can never be continuous. In addition to this compulsory discrete nature of the variables, an input space can be discrete according to the format of the input data representation.

If the format of an input variable is floating point, then the space between two consecutive points varies with the absolute values of these points. The discrete points are concentrated close to zero. The number of discrete points in a unit length decreases with the distance from the origin.

The structure of the input space is very complex when we use the floating point format for input variables. Hence for this study we analyze the application of domain testing in a discrete input space with uniform spacing Δ in all dimensions. This models the situation where the same data representation, integer or fixed-point, is used for all input variables.

In a discrete input space the distance from the OFF-point to the border segment does not depend on the machine tolerance. It depends on the distance of the closest discrete point from the centroid of the ON-points. Moreover, the probability of a vertex being a discrete input point is

small. Therefore, it is unlikely that a vertex will be chosen as an ON-point.

Another problem in a discrete space is that the concept of the actual border is ambiguous. There can be an infinite number of set of border segments which determine the same set of input points. Figure 4.1 illustrates the situation in 2-dimensions. Taking either ABCD or EFGHI as the boundary of the domain does not make any difference in program execution.

This problem makes it difficult to use BSE (the border shift error) as the error measure in a discrete space because this error measure does not represent the number of input points which may be in a wrong domain due to an undetectable error of the border. It is difficult to find a measure which represents the number of erroneous points. Since the BSE gives an upper bound for the error we shall continue to use it as the error measure.

4.2 Selecting Test Points in a Discrete Space.

4.2.1 ON-Points.

One disturbing problem with application of domain testing in a discrete space is that choosing N or (N+1) vertices on a border segment does not mean that we have found the required set of ON-points. We still have to find lattice points close to each selected vertex.

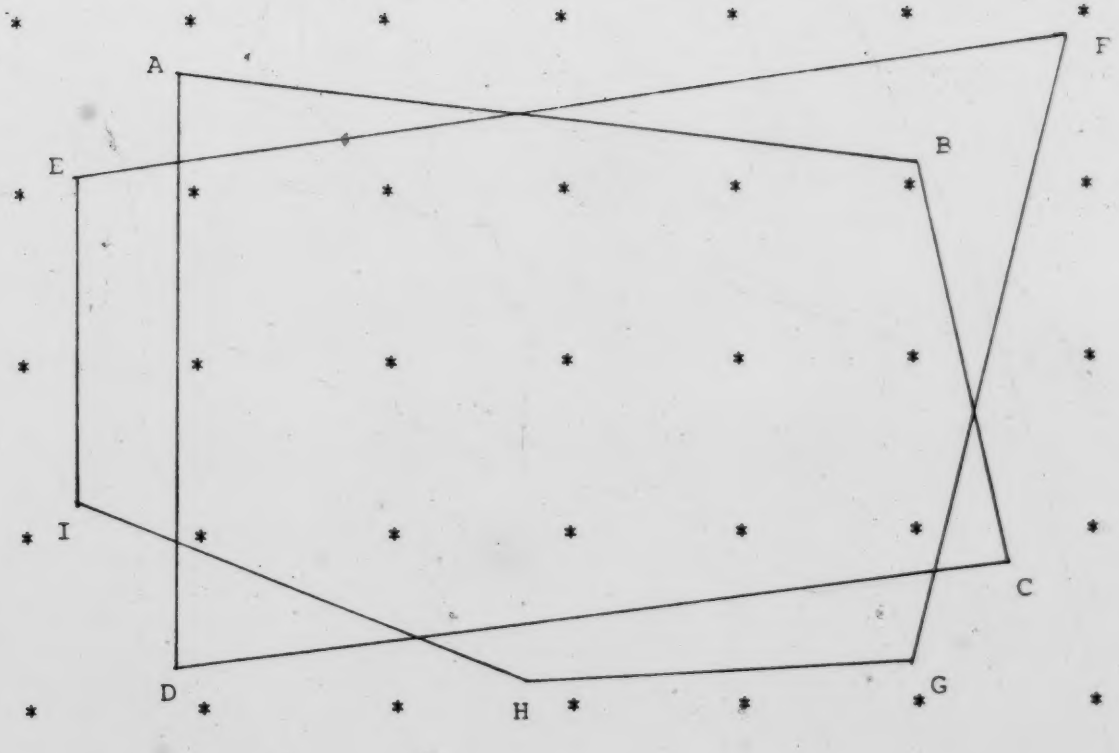


Figure 4.1 Correct Borders in 2-Dimensional Discrete Space

Since path conditions are linear, a straightforward method is to apply a linear integer programming technique to locate a discrete point in the domain. To reduce the error, ON-points should be selected as close as possible to the given border. When we apply the Scattering Method in a continuous space using the equality condition which corresponds to the given border we find ON-points on that border segment. Since there may be no lattice point on a border segment, we cannot use the equality condition in a discrete space. If we apply an integer programming technique under all the path conditions then the distance from the optimizing point to the given border can be very large and that point may not be desirable. If we use a linear integer programming technique to locate a lattice point close to a selected vertex, then we have to apply an optimization method twice for each ON-point; first to find the vertex and next to locate the lattice point. Because of these difficulties in applying integer programming, we consider possible alternative methods to find the discrete ON-points.

Using the following semicircle lemma, White et al. [14] have developed a method to locate a discrete test point close to a vertex, but not the closest, in 2-dimensions:

Lemma: Given a 2-dimensional discrete input space with resolution Δ, any semicircle with diameter $\sqrt{5}\Delta$ contains at least one lattice point.

Figure 4.2 shows how this lemma can be adapted to choose a discrete point close to a given vertex E. When the external angle $\theta \leq 90°$, as shown in Figure 4.2(a), the center S of the semicircle is given by $ES = \sqrt{5}\Delta/2$. Since the semicircle in this case can be drawn without regard for the given adjacent border, the location of the test point depends only on the vertex. If $\theta > 90°$, as shown in Figure 4.2(b),

$$ES = (\sqrt{5}\Delta/2)\ \mathrm{cosec}(\theta).$$

According to the semicircle lemma there exists at least one input point in the semicircle and we can use it as an ON-point. The distance from the given border segment to that point is at most $\sqrt{5}\Delta/2$.

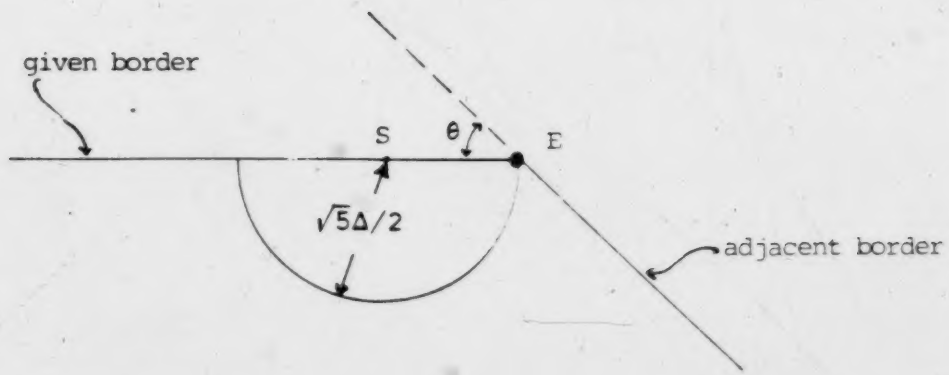
In the course of this research a better criterion has been developed. As shown in Figure 4.3, to test the given border b_1, we have to find an ON-point close to the vertex E. b_2 is the given adjacent border. Suppose d_1 and d_k are the distances between two borders along the Y direction at the first and k'th discrete lattice point lines along the X direction, then we have the relation

$$d_1/x_1 = d_k/(x_1+(k-1)\Delta)$$

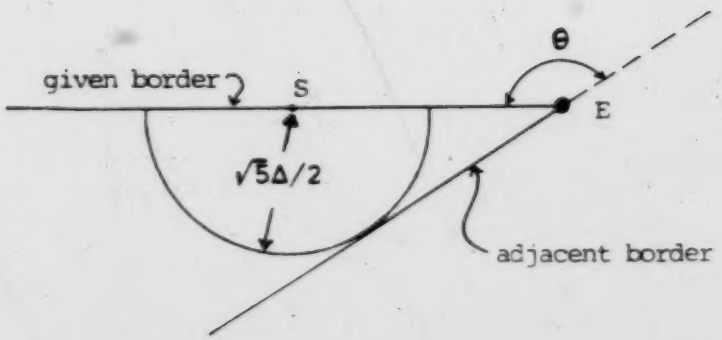
$$\text{i.e. } d_k = (x_1+(k-1)\Delta)d_1/x_1.$$

Since we know the equations of b_1 and b_2, we can calculate d_1.

When $d_k \geq \Delta$, we have at least one discrete point in the domain on that line. Using the above relation we can find the smallest k such that $d_k \geq \Delta$ and that k can be used to determine the distance from E in the X direction to a point



(a)



(b)

Figure 4.2 Selection of a Discrete ON-Point in the Semicircle Close to the Vertex E

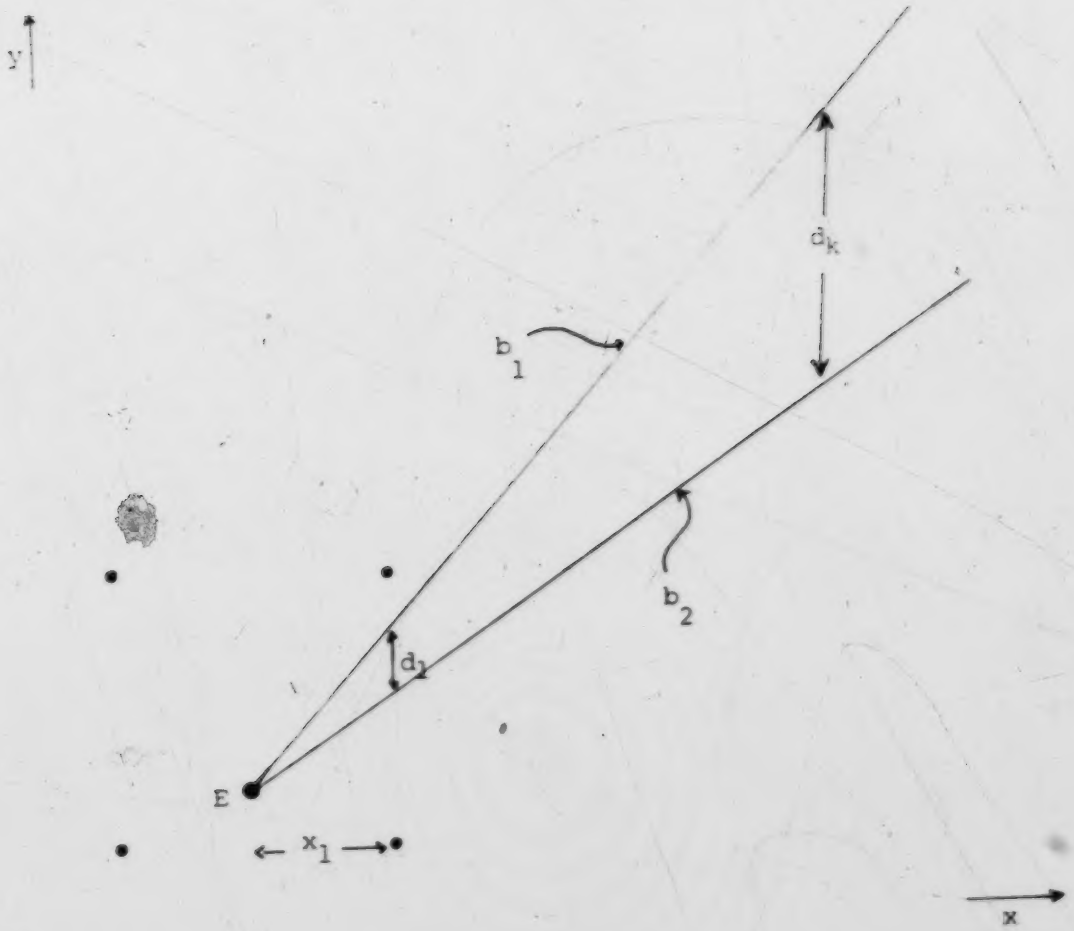


Figure 4.3 Position of a Discrete ON-Point Close to the Vertex E

where we can have at least one lattice point. In the same manner we can find the distance along the Y direction to a lattice line which guarantees a lattice point in the domain.

The coordinate direction which makes the largest angle with the centroid of b_1 and b_2 gives the smallest distance to the lattice line which guarantees at least one lattice point in the domain.

Since the distance to the given border from the ON-point selected by this method is at most Δ, this test point is more apt to be close to the given border segment than the test point selected by the semicircle method.

If the number of border segments through the vertex being considered is N, this method can be generalized to the N-dimensional case. To decide the dimension along which the search should be done, we find the dimension which makes the largest angle with the central axis of the the borders through the given border vertex E. Suppose this direction is x, and let P be the polyhedron defined by the predicates that correspond to the border segments through E. Then for each discrete value x_i of x, $x=x_i$ defines a lattice point hyperplane perpendicular to the x direction. Let x_1 be the smallest discrete value of x which is greater than the x coordinate (say a) of E. Without loss of generality, suppose $x=x_1$ intersects P and d_1 is the diameter of the largest (N-1)-dimensional sphere in the intersection of P and $x=x_1$. If x_k is the k'th discrete point of x from E then $x=x_k$ intersects P. Let d_k be the diameter of the largest

(N-1)-dimensional sphere in the intersection of P and $x=x_k$. If $x_1-a=r_1$ then we have the relation:

$$d_1/r_1 = d_k/(r_1+(k-1)\Delta).$$

Since an (N-1)-dimensional sphere with diameter $\sqrt{(N-1)}\Delta$ contains at least one lattice point, we can find the smallest distance along the x direction which guarantees at least one point in the domain by finding the smallest k which satisfies the above relation and $d_k \geq \sqrt{(N-1)}\Delta$. Note that to obtain a discrete point in the domain, it should be large enough to contain the corresponding (N-1)-dimensional sphere. By this method we can select an ON-point with a distance at most $\sqrt{(N-1)}\Delta$ from the given border.

The process of choosing a set of ON-points by this method requires first to find N or N+1 vertices and then to locate a discrete input point in the domain close to each selected vertex. A set of vertices can be found by the algorithm developed in the section 3.5 for the continuous space. The same algorithm can be used to select a set of ON-points directly without first finding a set of vertices. The N-dimensional sphere with diameter $\sqrt{N}\Delta$ contains at least one lattice point. Using this fact our algorithm has been developed to find N or N+1 discrete points close to the given border segment. Suppose we have to find a set of ON-points to test a border segment H which corresponds to the predicate $\beta_1^T X \geq b_1$. The predicate interpretation of the other path predicates are $\beta_k^T X \geq b_k$ for k=2,3,...,B, where B is the number of path predicates. Then H is determined by

$\beta_1^T X = b_1$ and the other path predicates. We consider the region defined by

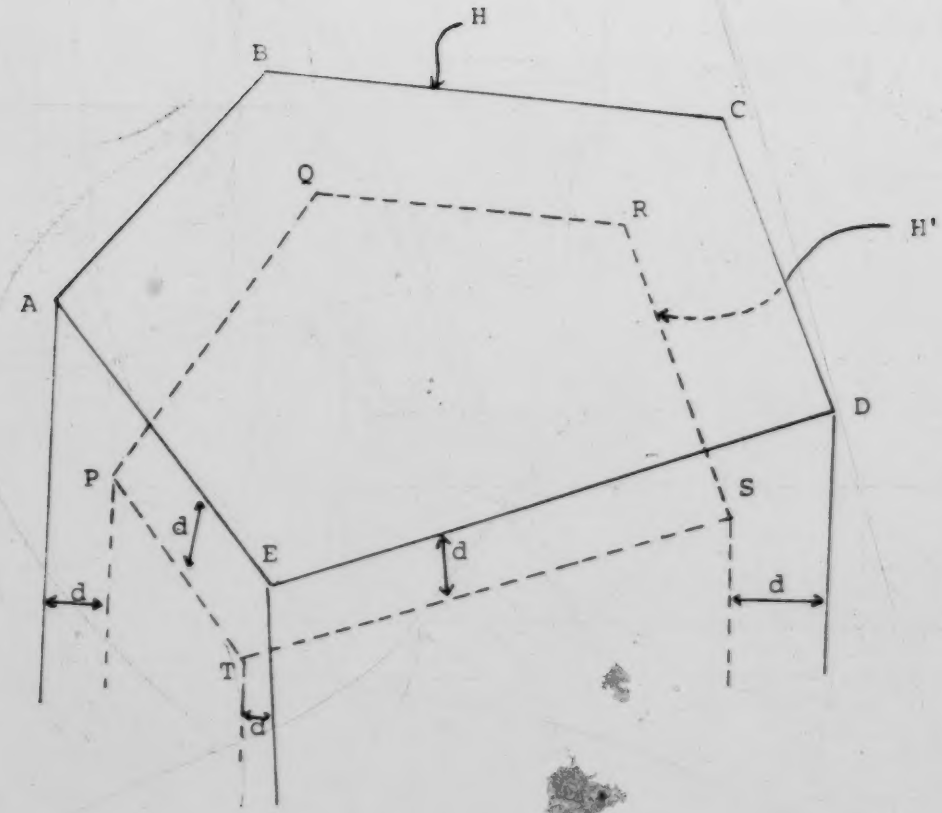
$$\beta_1^T X = b_1 + (N * \|\beta_1\|)^{1/2}\Delta/2 \text{ and}$$

$$\beta_k^T X \geq b_k + (N * \|\beta_k\|)^{1/2}\Delta/2 \text{ for } k=2,3,.....,B.$$

This region is a hyperplane segment (say H') which is parallel to the given border segment H separated by a distance $\sqrt{N}\Delta/2$. Also H', if it exists, is contained in the domain and the distance to a border segment from a point on H' is at least $\sqrt{N}\Delta/2$. For instance, consider the given border segment ABCDE in a 3-dimensional discrete input space depicted in Figure 4.4. Then PQRST is the subspace H' corresponding to the border segment ABCDE. Now using the above algorithm we can find a set of N or N+1 vertices of H'. Rounding off the coordinates of the selected vertices we can obtain a set of N or N+1 lattice points at a distance at most $\sqrt{N}\Delta$ from the given border segment.

If the given domain is very small, either H' may not exist or the dimension of H' may be less than N-1. Then it will be indicated by the failure of the linear programming algorithm. Such a border segment cannot be tested by this method.

We should note that for all the proposed methods, we first select a set of continuous points and then we locate a set of discrete points. Although those continuous points are linearly independent there is no guarantee that the discrete points are linearly independent. If we do not have a set of N linearly independent ON-points the the error can be



$d = \sqrt{3}\Delta/2$

Figure 4.4 Parallel Hyperplane Segment at Distance $\sqrt{N}\Delta/2$

unbounded. Therefore in a discrete space, we have to test for the the linear independence of ON-points after we select them.

4.2.2 OFF-Points.

To select the OFF-point such that the MAD is an acute angle, the projection of the OFF-point should be a convex combination of the projection points of the ON-points on the given border. A reasonable effort towards achieving this requirement is to find the OFF-point close to the centroid of the projection points of the ON-points on the given border. Rounding off the coordinates of the point which is $\sqrt{N}\Delta/2$ away from the centroid along the direction of the normal vector to the given border, we can obtain the OFF-point with less computational complexity.

If the tester intends to obtain better accuracy by selecting N or (N+1) OFF-points, then a set of OFF-points can be chosen by the same algorithm used to select ON-points. Since OFF-points should satisfy all path conditions except the condition which determines the given border segment, conditions for this case should be the converse condition of the given border segment, i.e. $\beta_i^T X < b_i$, but should satisfy all other path conditions.

4.3 Fundamental Limitations in a Discrete Space.

This section presents several cases in discrete space for which domain testing cannot be applied. These situations are characterized by the inability to find a sufficient number of test points because of the discrete nature of the input data.

Suppose the dimension of the given domain is N_d. Then at least N_d ON-points are required to test a border segment. This requirement cannot be achieved if the domain does not contain N_d or more input points. This can happen when either N_d is less than the dimension of the input space or the domain is too small to contain the required number of input points. A reduced dimensional domain occurs when path conditions contain one or more equality predicates. Moreover, two or more predicates can generate a condition equivalent to an equality predicate. For instance, two predicates $\beta^T X \geq b$ and $k\beta^T X \leq kb$, where k is a constant, are equivalent to the condition $\beta^T X = b$. If the domain is reduced in dimension, it may not contain any input points in a discrete space.

To understand the insufficient size of a domain let us define the *width* of a domain to be the smallest distance between parallel hyperplanes which contain the domain between them. If the width of the domain is on the order of the lattice resolution Δ, then it may not contain at least N_d input points. Two examples in 2-dimensions are shown in Figure 4.5. There is only one point in the domain ABC and

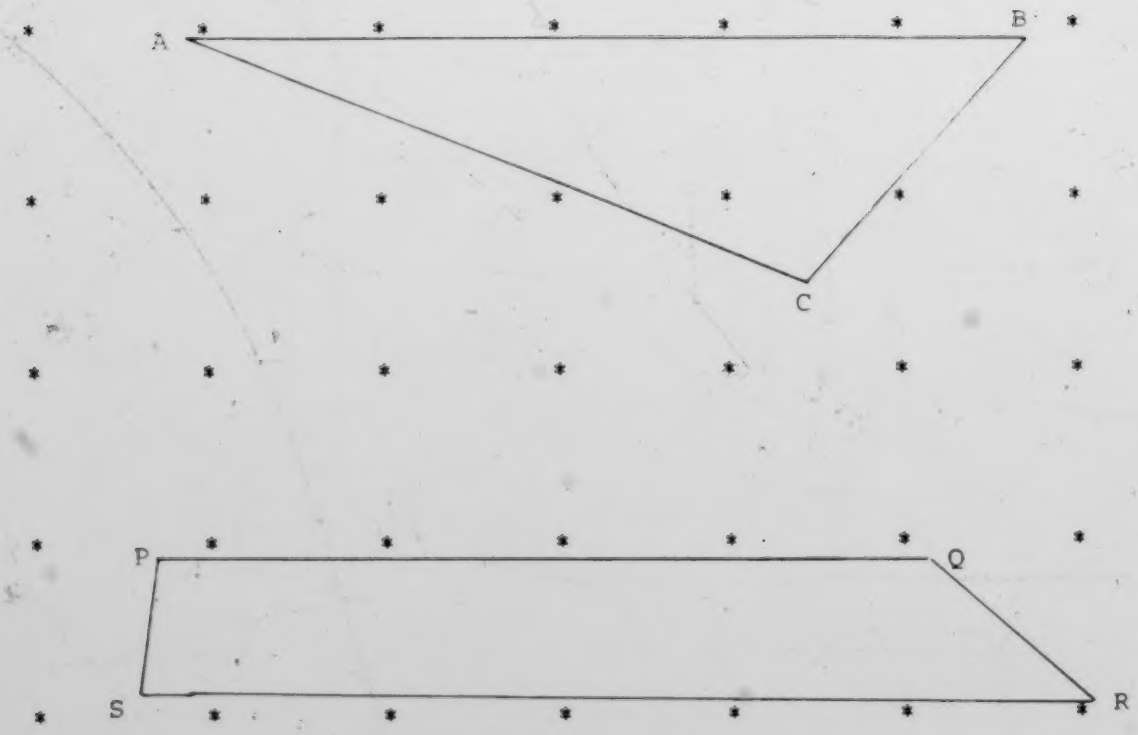


Figure 4.5 Examples for Domains with an Insufficient Width.

the domain PQRS contains no input points at all.

A border segment of a domain cannot be tested by the domain testing strategy if it is impossible to find an OFF test point. Since the OFF-point should satisfy all path conditions except that corresponding to the the given border, the existence of an OFF-point depends on the width of the polyhedron determined by those conditions and the converse condition of the given border. The N-dimensional sphere with diameter $\sqrt{N}\Delta$ contains at least one discrete point. Thus the existence of a sphere with diameter $\sqrt{N}\Delta$ in the polyhedron described above guarantees the existence of an OFF-point. Using this criterion, a sufficient condition for the existence of an OFF-point has been obtained for the 2-dimensional case [15]. It is obtained in terms of the *diameter* of the domain which is the shortest distance from any extreme point to a domain edge not adjacent to that extreme point, and the external angles of the given adjacent borders. The sufficient condition is formally stated as:

Given a domain with diameter d in a lattice with resolution Δ, a border segment EF and two given adjacent borders with external angles θ_1 and θ_2 if

$$d>(3/\sqrt{2})\Delta \quad \text{and} \quad |\cot(\theta_1)+\cot(\theta_2)| < EF/(3\Delta/\sqrt{2})$$

then a lattice OFF-point can be chosen to test the border segment EF.

When an external angle is very small (i.e. near 0°) this

condition may not be satisfied.

Another limitation of domain testing in discrete space occurs when there exists a set of test data to test a border segment but the error of the test is not within an acceptable limit. If the MAD of the tested border is an obtuse angle it cannot be considered as a reliable test. The MAD can be an obtuse angle if the projection of the OFF-point on the given border is not a convex combination of the projection points of ON-points on that border.

A sufficient condition for the finite BSE of a test for a border segment in a 2-dimensional lattice has been obtained by White et al. [14]. If the border segment is EF and the external angles of the given adjacent borders are θ_1 and θ_2 then a sufficient condition for a finite BSE is:

$$EF \geq (3\sqrt{5}\Delta/2)\{\text{cosec}(\theta_1) + \text{cosec}(\theta_2)\}$$

This condition is obtained by a geometrical analysis which involves the length of the border segment. Hence, it requires that both vertices are known. Although it is not difficult to find the external angles of the given adjacent borders in N-dimensional case, it is very difficult to find all the vertices. Moreover, a geometrical analysis in the N-dimensional case is not as simple as that for 2-dimensions. Therefore, it is difficult to obtain such sufficient conditions for a higher dimensional discrete input space. The reliability of a test in higher dimensions should be established by the MAD error analysis after selecting test points.

A sufficient condition for an acute MAD in 3-dimensional discrete space is obtained in Appendix C in terms of the width d of the parallelogram generated by the Scattering Method algorithm. If $d>6\sqrt{3}\Delta$ then the MAD of the test in 3-dimensions is an acute angle and

$$MAD \leq \tan^{-1}(12\sqrt{3}\Delta/(d-6\sqrt{3}\Delta)).$$

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Overview

The objective of this research is to examine, develop and analyze the possible methods of selecting test data for the domain testing strategy with an acceptable error bound. Major considerations are testing discrete and higher dimensional domains. The evaluation of techniques is done on the basis of the error bound, the number of test points and time required for the test.

Considering the role of representation of the potential error in domain testing, the BSE is found to be a good error measure since it models the maximum displaced domain that could result from an undetected border shift. Although the BSE can be manipulated easily in a 2-dimensional space, it is very complicated to analyze in higher dimensions. Thus, a new error measure, MAD, is introduced. Calculating the MAD in higher dimensions is not as complicated and it models a notion of the comparative size of the potential error as a portion of the entire input space. The major disadvantage of the MAD is its independence of the adjacent borders. Thus it fails to give an upper bound of the possible error as the BSE does. Nevertheless, we can expect a small BSE for a set of test data with a small MAD.

The necessary number of test points and the time required for a test is determined as a function of the

dimension N and the number of path predicates B. A low order polynomial complexity is considered reasonable.

We can increase the accuracy by testing subpath domain border segments instead of the path domain border segments. It does not increase the complexity of either the necessary number of test points or the time required for the test.

Three major testing methods proposed by [2,15] are examined on the basis of accuracy and complexity. The Nx1 [15] and NxN [2] strategies do not provide sufficient guidance to select test points. The best accuracy which can be obtained by the NxN strategy is better than that of the Nx1 method. Considering the accuracy, the VxV strategy cannot be worse than the NxN strategy, but it does not show a significant improvement. Although the number of test points in the Nx1 and NxN strategies is polynomial, the time complexity is nonpolynomial. The VxV strategy has nonpolynomial complexities in both the number of test points and the time.

An alternative strategy called the Scattering Method is proposed. This method does not guarantee the selection of the best set of test points, but it provides a good set with a small error. The pragmatic value of the Scattering Method is determined to be better than the other methods because it has a polynomial complexity of both time and the necessary number of test points. The effectiveness of this method can be improved by selecting an OFF point for each ON point.

In a discrete input space there are pathological situations for which domain testing cannot be done. Sufficient conditions to identify the cases with no finite error bound is obtained in 2-dimensional space. It is very difficult to obtain such conditions which can be applied before selecting test data in higher dimensions. The new criterion developed to select a discrete point close to a vertex in a 2-dimensional input space can be generalized to the N-dimensional case if the number of border segments through the vertex is N. The Scattering Method algorithm can be easily adapted to a discrete N-dimensional space, when the domain being tested is sufficiently large.

5.2 Contributions to Domain Testing

The major achievements of this research can be summarized as follows:

1. A new method has been developed to select ON test points for a border segment considering the subpath domain instead of the path domain. It increases the degree of confidence imparted by a successful test without increasing the complexity of the test.
2. Several results have been obtained in domain testing in higher dimensions. The most important achievement is to develop the Scattering Method to select test points. A mathematical approach to determine the finite BSE is developed. Due to the complexity of the domain structure and the BSE, a new error measure, the Maximum

Angular Displacement (MAD), is introduced.

3. The results obtained in a discrete space includes a new method to find a discrete point close to a given vertex of a domain. Moreover, the Scattering Method is adapted to obtain a set of discrete ON points close to a border segment without first finding a set of vertices.

5.3 Future Work

This research provides a theoretical analysis of the selection of test data for the Domain Testing Strategy. Future work should include experiments which compare the effectiveness and complexity of the Scattering Method, Nx1, NxN and VxV approaches for test selection. The appropriate measure of the effectiveness should be the MAD error criterion. Experiments should also be done for the discrete model, in order to evaluate the effect of the discrete space on test data selection.

We should note that among the existing methods the Scattering Method provides a set of input points close to the best in polynomial time. It is a requirement in most software production shops that programmers shall spend as little time as possible in testing. Therefore, testing methods with nonpolynomial time complexity are totally impractical. Further developments must be done to obtain better accuracy and lower order polynomial complexity.

Much work remains to be done in the application of domain testing in a discrete space. Most of the programs use

different formats to read different input variables. Real numbers can be read as floating point numbers. How are test points selected under these conditions? Another important problem to be considered in a discrete space is to develop an error measure which can represent the number of erroneous input points.

The domain testing strategies assume that the predicate interpretations corresponding to both the given and actual borders are linear. An initial modification to handle nonlinear predicates has been done by Zeil in [20]. One may work to improve this method by developing some practical guidelines to select test data for the predicates with low degree interpretations.

REFERENCES

1. Clarke L.A., "A System to Generate Test Data and Symbolically Execute Programs",IEEE Trans. Software Eng., vol. SE-2 pp.215-222, Sept. 1976.
2. Clarke L.A, Hassell J. and Richardson D.J., "A Close Look at Domain Testing", IEEE Trans. Software Eng., vol. SE-8 pp.380-390, July 1982.
3. DeMillo R.A., Lipton R.J. and Sayward F.G, "Hints on Test data Selection: Help for the Practing Programmer", Computer, vol.11, pp.34-43, April 1978.
4. Goodenough J.B. and Gerhart S.L., "Toward a Theory of Test Data Selection", IEEE Trans. Software Eng., vol.SE-1, pp.156-173, June 1975.
5. Grunbaum B., "Convex Polytopes", Interscience Publishers, John Wiley & Sons, London, 1967.
6. Howden W.E., "Methodology for the Generation of Program Test Data", IEEE Trans. Comput., vol.C-24, pp.554-560, May 1975.
7. Howden W.E., "Reliability of the Path Analysis Testing Strategy", IEEE Trans. Software Eng., vol.SE-3, pp.266-278, July 1977.
8. Jacobs W.W. and Schell E.D., "The number of vertices of a convex polytope", Amarican Math Monthly, vol.66, pp.643, 1959.
9. Klee V., "On the Number of Vertices of a Convex Polytope", Can. J. Math., Vol.16, pp.701-720, 1964.
10. Miller,Jr. E.F., "Program Testing:Guest Editor's

Introduction", Computer, vol.11, pp.10-12, April 1978.

11. Ramamoorthy C.V., Kim K.H. and Chen W.T., "On the Automated Generation of Program Test Data",IEEE Trans. Software Eng., vol.SE-1, pp.403-410, Dec.1975.
12. Tai K.C.,"Program Testing Complexity and Test Criteria", IEEE Trans. Software Eng., vol.SE-6, pp.531-538, Nov.1980.
13. Weyuker E.J. and Ostrand T.J.,"Theories of Program Testing and the Application of Revealing Subdomains", IEEE Trans. Software Eng., vol.SE-6, pp.236-246, May.1980.
14. White L.J., Teng F.C, Kuo H. and Coleman D., "An Error Analysis of Domain Testing Strategy", Technical report OSU-CISRC-TR-78-2, The Ohio State University, Columbus, Ohio.
15. White L.J., Cohen E.I. and Chandrasekaran B., "A Domain Strategy for Computer Program Testing", Technical report OSU-CISRC-TR-78-4, The Ohio State University, Columbus, Ohio.
16. White L.J. and Cohen E.I., "A Domain Strategy for Computer Program Testing", IEEE Trans. Software Eng., vol.SE-6, pp.247-257, May 1980.
17. White L.J. and Sahay P.B., "Experiments Determining Best Paths for Testing Computer Program Predicates", Technical Report TR 85-3, Department of Computing Science, The University of Alberta, Edmonton, Canada.
18. Yeh R.T. "Guest Editorial: Special Issue on Reliable

Software", Computing Surveys, vol.8, No.4, pp.355-356, Dec.1976.

19. Zeil S.J., "Selecting Sufficient Test Paths for Program Testing", Technical report OSU-CISRC-TR-81-10, The Ohio State University, Colombus, Ohio.

20. Zeil S.J., "Perturbation Testing for Domain Errors", COINS Technical report 83-38,Dec.1983, University of Massachusetts, Amherst, Massachusetts.

APPENDIX A

A Geometrical Interpretation of the Angle Between Two Hyperplanes.

Suppose two hyperplanes H_1 and H_2 in an N-dimensional space (N>1) are given by $\alpha^T X=a$ and $\beta^T X=b$ respectively. Here α and β are coefficient column vectors, X is a column vector in the space, a and b are constants. The angle between H_1 and H_2 is θ (>0). X_0 is a point on H_1, ϵ is the distance from X_0 to H_2 and r is the distance from the projection of X_0 on H_2 to the (N-2)-dimensional manifold generated by the intersection of H_1 and H_2.

Any hyperplane through the intersection of H_1 and H_2 is given by,

$$(t\alpha+(1-t)\beta)^T X = t*a+(1-t)b_2$$

where t is a constant.

Suppose $(t_1\alpha+(1-t_1)\beta)^T X = t_1*a+(1-t_1)b$ is perpendicular to H_2. Then,

$$(t_1\alpha+(1-t_1)\beta).\beta = 0$$

$$\text{i.e. } t_1 = \|\beta\|^2/(\|\beta\|^2-\|\alpha\|\|\beta\|\cos\theta) \quad \text{and}$$

$$1-t_1 = -\|\alpha\|\|\beta\|\cos\theta/(\|\beta\|^2-\|\alpha\|\|\beta\|\cos\theta)$$

∴ The hyperplane perpendicular to H_2 through the intersection of H_1 and H_2 is,

$$(\|\beta\|^2\alpha-(\|\alpha\|\|\beta\|\cos\theta)\beta)^T X = |\beta|^2a-(|\alpha\|\|\beta\|\cos\theta)b_2$$

Let's take this equation as $\gamma^T X=c$. Then r is the distance to $\gamma^T X=c$ from X_0.

$$\therefore r = (|\gamma^T X_0-c|)/\|\gamma\|$$

$$\gamma^T X_0 = \|\beta\|^2 a^T X_0 - (\|a\|\|\beta\|\cos\theta)\beta^T X_0$$

Since X_0 is on H_1, $a^T X_0 = a$.

$$\therefore \gamma^T X_0 = \|\beta\|^2 a - (\|a\|\|\beta\|\cos\theta)\beta^T X_0$$

$$\therefore |\gamma^T X_0 - c| = \|a\|\|\beta\|\cos\theta\,|\beta^T X_0 - b|$$

$$\epsilon = (|\beta^T X_0 - b|)/\|\beta\|$$

$$\therefore \epsilon/r = \|\gamma\|/\|a\|\|\beta\|^2\cos\theta$$

$$\|\gamma\|^2 = \|(\|\beta\|^2 a - (\|a\|\|\beta\|\cos\theta)\beta)\|^2$$

$$= \|\beta\|^4\|a\|^2 + \|a\|^2\|\beta\|^2\cos^2\theta\|\beta\|^2 - 2\|\beta\|^2\|a\|\|\beta\|\cos\theta\|a\|\|\beta\|\cos\theta$$

$$= \|a\|^2\|\beta\|^4(1-\cos^2\theta)$$

$$= \|a\|^2\|\beta\|^4\sin^2\theta$$

$$\therefore \|\gamma\| = \|a\|\|\beta\|^2\sin\theta$$

$$\therefore \epsilon/r = \sin\theta/\cos\theta = \tan\theta$$

Therefore, the angle between H_1 and H_2 is given by $\tan^{-1}(\epsilon/r)$.

APPENDIX B

An Upper Bound for the MAD in 3-Dimensions

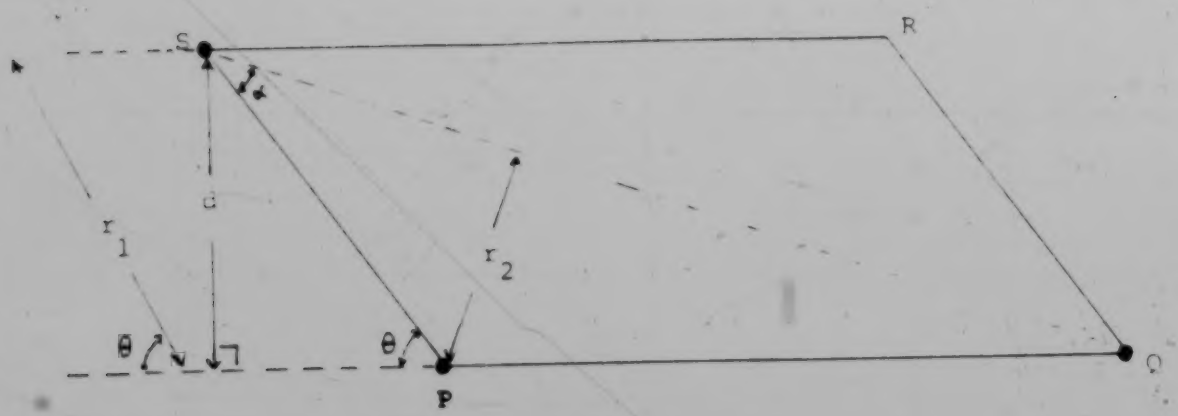


Figure B.1

The MAD is determined by the angle between the given plane and the plane determined by S, Q and the OFF point. Suppose the OFF point is selected at ϵ distance from the centroid of P, Q and S, which is at the 1/3 of the distance from QS to P.

$$r_1 = d/\sin(\theta) = d/(2*\sin(\theta/2)*\cos(\theta/2))$$

$$r_2 = r_1\sin(\alpha) = d*\sin(\alpha)/2*\sin(\theta/2)*\cos(\theta/2)$$

Since d is the minimum of the distances between parallel lines $\alpha \geq \theta/2$ and $\sin\alpha \geq \sin(\theta/2)$.

$$\therefore r_2 \geq (d*\sec(\theta/2))/2 \geq d/2.$$

$$\therefore \text{MAD} \leq \tan^{-1}(\epsilon/(r_2/3)) = \tan^{-1}(6\epsilon/d)$$

APPENDIX C

A Sufficient Condition for an Acute MAD in 3-Dimensional Discrete Space

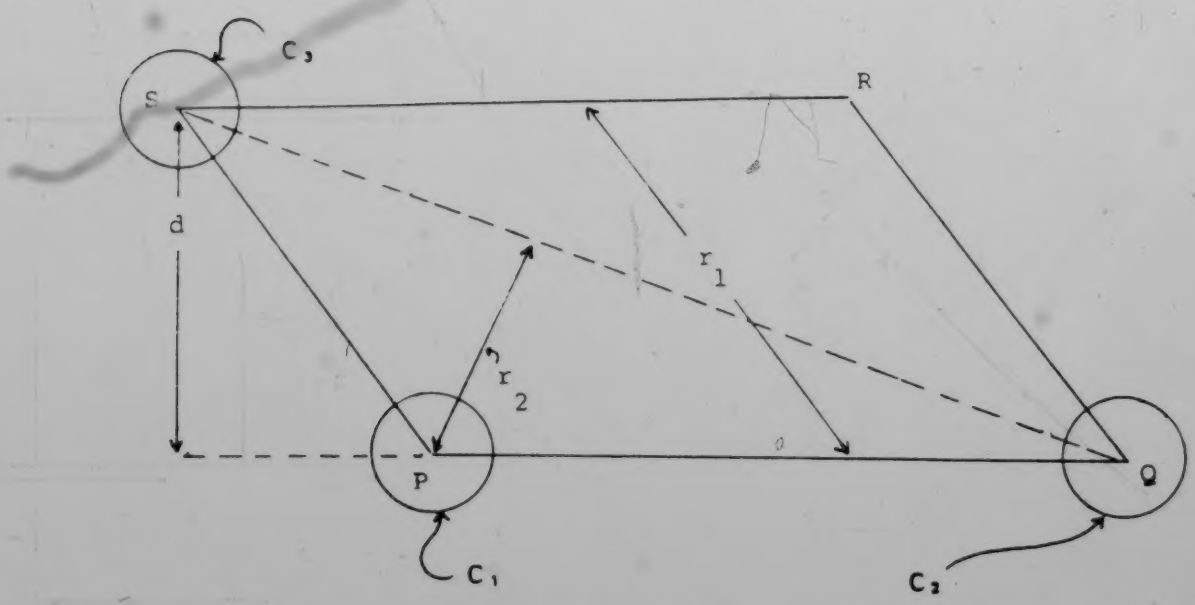


Figure C.1

A sufficient condition which guarantees that the MAD is an acute angle in 3-dimensional space can be obtained in terms of the smallest distance between the parallel lines generated by the Scattering Method algorithm. Unlike a continuous space, for this case the parallelogram is obtained on a plane at distance $\sqrt{3}\Delta/2$ from the given border. The worst case occurs when the selected points are three of the vertices of the parallelogram. As shown in Figure C.1, if the points selected by the algorithm are P,Q and S, then the lattice points obtained by rounding off their

coordinates must be three points at most $\sqrt{3}\Delta/2$ away from P, Q and S respectively. Suppose they are P', Q' and S'.

The projection points on these three lattice points on the parallelogram lie in the circles C_1, C_2 and C_3. The OFF point is selected by rounding the cordinates of the point which is $\sqrt{3}\Delta$ away from the centroid of P,Q and S. Therefore, the distance x parallel to the given plane from the OFF point to Q'S' is,

$$x \geq r_2/3 - \sqrt{3}\Delta$$

Since $r_2 \geq d/2$, $x \geq d/6 - \sqrt{3}\Delta = (d-6\sqrt{3}\Delta)/6$.

The perpendicular distance y is,

$$y \leq 2\sqrt{3}\Delta$$

$\therefore \tan(MAD) = y/x \leq 12\sqrt{3}\Delta/(d-6\sqrt{3}\Delta)$

Therefore if $d > 6\sqrt{3}\Delta$ then it guarantees that the MAD is an acute angle.